Zero-Dimensional Gröbner Bases for Rescue-XLIX

Matthias Johann Steiner

Alpen-Adria-Universität Klagenfurt, Klagenfurt am Wörthersee, Austria
matthias.steiner@aau.at

Abstract. Rescue-XLIX is an Arithmetization-Oriented Substitution-Permutation Network over prime fields $\mathbb{F}_p$ which in one full round first applies a SPN based on $x \mapsto x^d$ followed by a SPN based on the inverse power map $x \mapsto x^{\frac{1}{d}}$. In a recent work, zero-dimensional Gröbner bases for SPN and Poseidon sponge functions have been constructed by utilizing weight orders. Following this approach we construct zero-dimensional Gröbner bases for Rescue-XLIX ciphers and sponge functions.

Keywords: Gröbner basis · Sponge function · Substitution-Permutation Network · Rescue-XLIX · Rescue-Prime

1 Introduction

Rescue [AAB⁺20] is a family of *Arithmetization-Oriented* (AO) ciphers and hash functions targeted for *Multi-Party Computation* (MPC) and *Zero-Knowledge* (ZK) applications. Rescue follows the classical Substitution-Permutation Network (SPN) design strategy, but it is only specified over prime fields $\mathbb{F}_p$. One full round of Rescue first applies a SPN based on the inverse power permutation $x \mapsto x^{\frac{1}{d}}$ followed by a SPN based on the power permutation $x \mapsto x^d$. Rescue-XLIX[1] [SAD20] is a standardization of the Rescue strategy which swaps the order of the SPNs, i.e. it first applies $x \mapsto x^d$ followed by $x \mapsto x^{\frac{1}{d}}$. The hash function derived from the Rescue-XLIX permutation is called Rescue-Prime. In this paper, to have a clear terminology, we denote the hash function derived from the Rescue permutation as Rescue-Prime, and the hash function derived from the Rescue-XLIX permutation as Rescue-XLIX-Prime respectively. Main aim of the Rescue strategy was to design a primitive for MPC & ZK applications with a strong security argument and decent efficiency with respect to the target metrics in applications [AAB⁺20, §4.1]. We note that the Rescue family has seen various cryptanalysis since its inception [FP19, BCL⁺20, BCP23].

A common feature of AO designs is to admit low degree iterated polynomial models. Hence, a lot of attention in AO cryptanalysis is given to polynomial system solving techniques, in particular to *Gröbner bases*. A Gröbner basis attack is typically divided into four steps:

(1) Model the cryptographic function of interest via a zero-dimensional polynomial system.

(2) Compute a Gröbner basis with respect to an efficient term order, typically one chooses the *degree reverse lexicographic* (DRL) term order.

(3) Perform term order conversion to an elimination order, typically one chooses the *lexicographic* (LEX) term order.

(4) Factor the univariate polynomial

[1] Pronounced as Rescue Forty Nine.

Note that for complexity estimations of Steps (3) and (4) one requires combinatorial knowledge about any Gröbner basis of the polynomial system. Since this knowledge a priori is not available one usually finds a phrase like

> *we base the security of our design on the hardness*
> *of computing a DRL Gröbner basis*

to justify resistance against Gröbner basis attack, see e.g. the `Rescue` design [AAB+20, §4.2.3]. Unfortunately, provable DRL Gröbner basis complexity estimation is a non-trivial problem. Recent progress was made by Steiner [Ste24a] who proved such estimations for the `MiMC` [AGR+16], `GMiMC` [AGP+19] and Hades [GLR+20] families, but he also provided evidence that his proving technique will fail for SPN sponge functions [Ste24a, §6.3]. Therefore, cryptographic designers usually resort to unproven hypotheses, like being *(cryptographically) semi-regular* [BFS04, BDND+21], and extrapolations of small scale experiments, see e.g. [AAB+20, §6.1].

In another recent work, Steiner [Ste24b] trivialized Step (2) for Poseidon [GKR+21] by constructing zero-dimensional Gröbner bases for preimage as well as CICO polynomial systems with respect to a weight order. In this work, we extend this approach to `Rescue-XLIX`. In particular, we construct Gröbner bases with respect to weight orders for:

(I) `Rescue-XLIX` ciphers with affine and non-affine key schedules (Theorem 3.1).

(II) `Rescue-XLIX-Prime` preimage polynomial systems (Section 3.2.1).

(III) `Rescue-XLIX-Prime` CICO polynomial system (Section 3.2.2).

We restrict our analysis to `Rescue-XLIX`, because its iterated polynomial model is slightly simpler than the one for `Rescue`. Though, we expect that the techniques developed in this paper can be generalized to `Rescue`.

For completeness, affine key schedules have not been proposed for `Rescue` or `Rescue-XLIX`. We study them to quantify the trade-off between ciphers and sponge functions, and to compare the trade-off to Hades & Poseidon.

Let r denote the number of full rounds of a `Rescue-XLIX` instance. As our main result, the $\mathbb{F}_q$-vector space dimension of `Rescue-XLIX` ciphers is given by

$$D_{\texttt{Rescue-XLIX}} = \begin{cases} d^r, & \text{affine key schedule,} \\ d^{2 \cdot r}, & \text{non-affine key schedule,} \end{cases} \tag{1}$$

and for `Rescue-XLIX-Prime` preimage as well as CICO polynomial systems the dimension is given by

$$D_{\texttt{Rescue-XLIX-Prime}} = d^r. \tag{2}$$

In particular, for `Rescue-XLIX-Prime` the $\mathbb{F}_q$-vector space dimension is invariant for the rate of the sponge.

1.1 Related Works

Before investigating Poseidon, Steiner [Ste24b] constructed Gröbner bases for SPN sponge functions. In principle, we follow the approach outlined by Steiner for `Rescue-XLIX-Prime`, but we can exploit a unique feature of `Rescue-XLIX` to simplify the construction. For `Rescue-XLIX-Prime` polynomial systems, in the polynomials representing one half round all non-constant terms have the same degree, i.e. the only non-homogeneous terms are constants. In the SPN weight order, the weights had to be iteratively increased, see [Ste24b, §3.1], but due to the unique feature of `Rescue-XLIX` polynomial models we only have to work with the weights 0 and 1.

Moreover, Steiner constructed a DRL Gröbner basis for SPN ciphers [Ste24a, Theorem 6.2]. His proof can be generalized to any affine key schedule and any combination of full and partial Substitution Layers as long as the first SPN has a full Layer. In particular, his proof yields a DRL Gröbner basis for Hades. On the other hand, he also pointed out that his technique will fail for non-affine key schedules [Ste24a, §6.4]. For `Rescue-XLIX` we resolve the problem for non-affine key schedules by considering a weight order instead of the standard DRL term order.

1.2 Organization of the Paper

In Section 2 we recall the mathematical requirements for this paper, and we formally introduce the sponge construction and the `Rescue` family.

In Section 3 we construct our `Rescue-XLIX` Gröbner bases. We start with the `Rescue-XLIX` cipher (Section 3.1), then we study preimage polynomial system for `Rescue-XLIX-Prime` (Section 3.2.1), and we finish by extending preimage Gröbner bases to CICO polynomial systems (Section 3.2.2)

In Section 4 we discuss the cryptanalytic impact of our Gröbner bases, and finally we finish with a short discussion in Section 5.

2 Preliminaries

Let q be a prime power, we denote the finite field with q elements by $\mathbb{F}_q$. We denote matrices $\mathbf{M} \in \mathbb{F}_q^{m \times n}$ with bold capital letters and vectors $\mathbf{v} \in \mathbb{F}_q^n$ with bold lower letters. Matrix-vector products are denoted as $\mathbf{Mv}$ and analog for matrix-matrix products.

Let $k \leq n$ be integers, and let $\mathbf{v} = (v_1, \dots, v_n)^\intercal \in \mathbb{F}_q^n$. We denote with $\mathbf{v}|^k = (v_1, \dots, v_k)^\intercal$ the truncation to its first k elements, and by $\mathbf{v}|_k = (v_{n-k}, \dots, v_n)^\intercal$ the restriction to its last k elements.

We denote with $\mathbf{I}_{m \times n} \in \mathbb{F}_q^{m \times n}$ the identity matrix, and with $\mathbf{0}_{m \times n} \in \mathbb{F}_q^{m \times n}$ the zero matrix. Also, we denote $\mathbf{1}_n = (1, \dots, 1)^\intercal \in \mathbb{F}_q^n$ and $\mathbf{0}_n = (0, \dots, 0)^\intercal \in \mathbb{F}_q^n$.

We denote the standard inner product of vectors as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\intercal \mathbf{y} = \sum_{i=1}^{n} x_i \cdot y_i. \tag{3}$$

The natural logarithm will be denoted as $\log(x)$ and logarithms in base b as $\log_b(x)$.

2.1 Sponge Construction

With the sponge construction [BDPV07, BDPV08] one can transform arbitrary functions, in particular cryptographic permutations, into a function which can digest arbitrary length inputs and can construct arbitrary length outputs. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $n = r + c$, where $n, r, c \in \mathbb{Z}_{\geq 1}$. One splits the input of f into r *rate* and c *capacity* bits. To digest a finite message $m \in \mathbb{F}_2^*$ it is split into $m = (m_1, \dots, m_N)$, where $m_i \in \mathbb{F}_2^r$ for all N. (If necessary, one has to pad the message m so that it has length $r \cdot N$.) Then we evaluate $f(m_1, \texttt{IV})$, where $\texttt{IV} \in F_2^c$ is some deterministic initial value, next we evaluate $f\big((m_2, \mathbf{0}) + f(m_1, \texttt{IV})\big)$, and this procedure is iterated until all message blocks have been digested. As hash value one returns the first r bits of the result of the digestion, if more output bits are required one calls f another time and again returns the first r bits. In Figure 1 we provide a visual representation of the sponge construction.

Most famous example for a sponge hash function is `Keccak` [BDPA13] which has been selected in the third iteration of NIST's *Secure Hashing Algorithm* standardization (SHA-3).

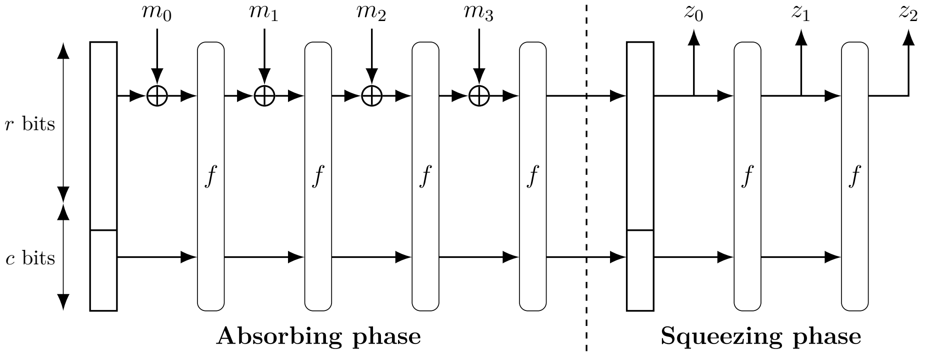

Figure 1: Visualization of the sponge construction, figure by [Jea16].

Over prime fields, one cannot divide the output of a function into bits and simultaneously be compatible with the field structure. Therefore, the sponge construction must be modified a bit.

Definition 2.1 ([Ste24b, Definition 2.1])**.** *Let $\mathbb{F}_q$ be a finite field, let $n, r_{in}, r_{out}, c \in \mathbb{Z}_{\geq 1}$ be such that $n = r_{in} + c$ and $r_{out} < n$, and let $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ be a function. Let $\mathtt{IV} \in \mathbb{F}_q^c$ be an initial value, and let $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_k) \in \mathbb{F}_q^{r_{in} \cdot k}$ be a message such that $\mathbf{m}_i \in \mathbb{F}_q^{r_{in}}$ for all i. To digest $\mathbf{m}$ via f in sponge mode one iterates through:*

(1) $\mathbf{y}_1 = f(\mathbf{m}_1, \mathtt{IV})$.

(2) For $2 \leq i \leq k$, $\mathbf{y}_i = f\big((\mathbf{m}_i, \mathbf{0}_c)^\intercal + \mathbf{y}_{i-1}\big)$.

To return an output in $\mathbb{F}_q^{(n-r_{out})\cdot l}$ *one iterates through:*

(1) $\mathbf{z}_1 = \mathbf{y}_k|^{n-r_{out}}$.

(2) For $2 \leq i \leq l$, $\mathbf{y}_{k+i} = f(\mathbf{y}_{k+i-1})$ *and* $\mathbf{z}_i = \mathbf{y}_{k+i}|^{n-r_{out}}$.

(3) Return $(\mathbf{z}_1, \ldots, \mathbf{z}_l)$.

From now on we will always denote with r_{in} the input rate of a sponge function and with r_{out} the "output rate" of the sponge, i.e. the size of the truncated output.

For AO hash functions, the sponge construction is a popular choice to construct hash functions, e.g. `Rescue-Prime` & `Rescue-XLIX-Prime` [AAB+20, SAD20], Poseidon & Poseidon2 [GKR+21, GKS23], `GMiMC` [AGP+19], `Anemoi` [BBC+23] and Griffin [GHR+23].

2.1.1 Computational Problems for Sponge Functions

In this paper we will investigate polynomial systems for *preimage* and *Constrained-Input Constrained-Output* (CICO) [BDPV11, §8.2.4] problems of sponge functions. For a preimage problem, one is given an initial value $\boldsymbol{\alpha} \in \mathbb{F}_q^{n-r_{in}}$ and a hash value $\boldsymbol{\beta} \in \mathbb{F}_q^{n-r_{out}}$, then one asks for a solution to the equation

$$f\begin{pmatrix} \mathbf{x}_{in} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta} \\ \mathbf{x}_{out} \end{pmatrix}, \tag{4}$$

where $\mathbf{x}_{in} = (x_{in,1}, \ldots, x_{in,r_{in}})^\intercal$ and $\mathbf{x}_{out} = (x_{out,1}, \ldots, x_{out,r_{out}})^\intercal$ are variables.

For a CICO problem, one is given two constants $\boldsymbol{\alpha} \in \mathbb{F}_q^{n-r_{in}}$ and $\boldsymbol{\beta} \in \mathbb{F}_q^{n-r_{out}}$, then one asks for a solution to the equation

$$f\begin{pmatrix} \mathbf{x}_{in} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{out} \\ \boldsymbol{\beta} \end{pmatrix}. \tag{5}$$

Note that these problems are only fully determined if $r_{in} + r_{out} \leq n$.

A third problem crucial for the security of hash functions is the so-called *collision* problem. Let $\boldsymbol{\alpha} \in \mathbb{F}_q^{n-r_{in}}$ and $\boldsymbol{\beta} \in \mathbb{F}_q^{n-r_{in}}$ be initial values, then one asks for a solution to the equation

$$f\begin{pmatrix} \mathbf{x}_{in} \\ \boldsymbol{\alpha} \end{pmatrix} = f\begin{pmatrix} \mathbf{y}_{in} \\ \boldsymbol{\alpha} \end{pmatrix}, \tag{6}$$

where $\mathbf{x}_{in} = (x_{in,1}, \dots, x_{in,r_{in}})^\intercal$ and $\mathbf{y}_{in} = (y_{in,1}, \dots, y_{in,r_{in}})^\intercal$ are variables.

An algorithm that solves one of these problems efficiently undermines the security of a sponge function.

2.2 The Rescue Family

`Rescue` & `Rescue-XLIX` [AAB+20, SAD20, AKM+22] are (keyed) cryptographic permutations targeted for MPC & ZK applications. Let us now formally recall their definitions.

Definition 2.2 (`Rescue` & `Rescue-XLIX`)**.** *Let $\mathbb{F}_q$ be a finite field, let $d, n, r \in \mathbb{Z}_{\geq 1}^n$ be such that $\gcd(d, q-1) = 1$, let $\mathbf{M}_0, \dots, \mathbf{M}_{2\cdot r} \in \mathbb{F}_q^n$ be invertible matrices, and let $\mathbf{c}_1, \dots, \mathbf{c}_{2\cdot r} \in \mathbb{F}_q^n$ be constants.*

(1) The Substitution Layer is defined as

$$\begin{aligned} \mathcal{S}_d : \mathbb{F}_q^n &\to \mathbb{F}_q^n, \\ (x_1, \dots, x_n)^\intercal &\mapsto \left(x_1^d, \dots, x_n^d\right)^\intercal. \end{aligned}$$

(2) For $1 \leq i \leq r$, the i^{th} keyed Substitution-Permutation Network is defined as

$$\begin{aligned} \mathcal{R}_d^{(i)} : \mathbb{F}_q^n \times \mathbb{F}_q^n &\to \mathbb{F}_q^n, \\ (\mathbf{x}, \mathbf{y}) &\mapsto \mathbf{M}_i \mathcal{S}_d(\mathbf{x}) + \mathbf{y} + \mathbf{c}_i. \end{aligned}$$

(3) The `Rescue` cipher is defined as

$$\begin{aligned} \texttt{Rescue} : \mathbb{F}_q^n \times \mathbb{F}_q^{(r+1)\cdot n} &\to \mathbb{F}_q^n, \\ (\mathbf{x}, \mathbf{y}_0, \dots, \mathbf{y}_{2\cdot r}) &\mapsto \mathcal{R}_d^{(2\cdot r)} \circ \mathcal{R}_{\frac{1}{d}}^{(2\cdot r-1)} \circ \dots \circ \mathcal{R}_d^{(2)} \circ \mathcal{R}_{\frac{1}{d}}^{(1)}\big(\mathbf{M}_0 \cdot (\mathbf{x} + \mathbf{y}_0)\big), \end{aligned}$$

where composition is taken with respect to the plaintext variable $\mathbf{x}$.

(4) The `Rescue-XLIX` cipher is defined as

$$\begin{aligned} \texttt{Rescue} : \mathbb{F}_q^n \times \mathbb{F}_q^{(r+1)\cdot n} &\to \mathbb{F}_q^n, \\ (\mathbf{x}, \mathbf{y}_0, \dots, \mathbf{y}_{2\cdot r}) &\mapsto \mathcal{R}_{\frac{1}{d}}^{(2\cdot r)} \circ \mathcal{R}_d^{(2\cdot r-1)} \circ \dots \circ \mathcal{R}_{\frac{1}{d}}^{(2)} \circ \mathcal{R}_d^{(1)}\big(\mathbf{M}_0 \cdot (\mathbf{x} + \mathbf{y}_0)\big), \end{aligned}$$

where composition is taken with respect to the plaintext variable.

Remark 2.3. It is well-known that an integer $d \in \mathbb{Z}$ induces a power permutation $x \mapsto x^d$ over $\mathbb{F}_q$ if and only if $\gcd(d, q-1) = 1$, see [LN97, 7.8. Theorem].

Note that the only difference between `Rescue` & `Rescue-XLIX` is the order of x^d and $x^{\frac{1}{d}}$ in the round functions. In Figure 2 we provide an illustration of the `Rescue-XLIX` round function, the SPNs for x^d and $x^{\frac{1}{d}}$ are called a half round, and their composition is called a full round.

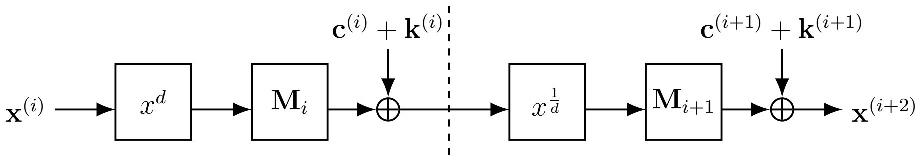

Figure 2: Illustration of a full round of `Rescue-XLIX` with key addition.

To obtain a hash function the designers proposed to instantiate `Rescue` & `Rescue-XLIX` in sponge mode. Moreover, in [SAD20] the `Rescue-XLIX` sponge function was baptized `Rescue-Prime`. To have a clear separation between these sponge functions, in this paper we denote with `Rescue-Prime` the sponge function of `Rescue` and with `Rescue-XLIX-Prime` the sponge function of `Rescue-XLIX`.

For the matrices, the designers proposed to use *Maximum Distance Separable* (MDS) matrices, see [AAB+20, §6]. They also provide a `SageMath` [Sag23] tool[2] to compute round numbers and generate round constants as well as an MDS matrix.

In this paper, we study two key schedules:

- Affine key schedules, i.e. for all $1 \leq i \leq 2 \cdot r$

$$\mathbf{y}_i = \hat{\mathbf{M}}_i \mathbf{y}_{i-1} + \mathbf{d}_i, \tag{7}$$

  where $\hat{\mathbf{M}}_i \in \mathbb{F}_q^{n \times n}$ and $\mathbf{d}_i \in \mathbb{F}_q^n$.

- Non-affine key schedules via the `Rescue-XLIX` round function, i.e. for all $1 \leq i \leq 2 \cdot r$

$$\mathbf{y}_i = \begin{cases} \mathbf{M}_i \mathcal{S}_d \left(\mathbf{y}_{i-1} \right) + \mathbf{d}_i, & \left\{ \begin{matrix} i \equiv 0 \mod 2 \text{ for } \texttt{Rescue}, \\ i \equiv 1 \mod 2 \text{ for } \texttt{Rescue-XLIX} \end{matrix} \right\}, \\ \mathbf{M}_i \mathcal{S}_{\frac{1}{d}} \left(\mathbf{y}_{i-1} \right) + \mathbf{d}_i, & \left\{ \begin{matrix} i \equiv 1 \mod 2 \text{ for } \texttt{Rescue}, \\ i \equiv 0 \mod 2 \text{ for } \texttt{Rescue-XLIX} \end{matrix} \right\}, \end{cases} \tag{8}$$

  where $\mathbf{d}_i \in \mathbb{F}_q^n$.

The latter key schedule is the standard key schedule for `Rescue` & `Rescue-XLIX`.

Before we formally define the `Rescue-XLIX` polynomial models, let us first illustrate how we set up an efficient model for `Rescue-XLIX`. Let $\mathbf{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)} \right)^\intercal$, $\mathbf{y}^{(i)} = \left(y_1^{(i)}, \ldots, y_n^{(i)} \right)^\intercal$ and $\mathbf{z}^{(i+1)} = \left(z_1^{(i+1)}, \ldots, \mathbf{z}_n^{(i+1)} \right)^\intercal$ be variables, where we consider $\mathbf{z}^{(i+1)}$ as auxiliary variables. Let i be odd, then for the i^{th} round we consider the equations

$$\mathbf{M}_i \mathcal{S}_d \left(\mathbf{x}^{(i)} \right) + \mathbf{y}^{(i)} + \mathbf{c}_i = \mathbf{x}^{(i+1)}, \tag{9}$$

for the inverse S-box $x^{\frac{1}{d}}$ we consider the auxiliary equations

$$\mathbf{x}^{(i+1)} = \mathcal{S}_d \left(\mathbf{z}^{(i+1)} \right), \tag{10}$$

and for $(i+1)^{\text{th}}$ round we consider the equation

$$\mathbf{M}_{i+1} \mathbf{z}^{(i+1)} + \mathbf{y}^{(i+1)} + \mathbf{c}_{i+1} = \mathbf{x}^{(i+2)}. \tag{11}$$

Note that for $\mathbb{F}_q$-valued solutions Equation (10) is equivalent to

$$\mathcal{S}_{\frac{1}{d}} \left(\mathbf{x}^{(i+1)} \right) = \mathbf{z}^{(i+1)}. \tag{12}$$

[2] https://github.com/KULeuven-COSIC/Marvellous

We can substitute Equation (10) into Equation (9) to omit the auxiliary equations. Also, we can then rename the auxiliary variables $\mathbf{z}^{(i+1)}$ as $\mathbf{x}^{(i+1)}$ to have an uniform notation. An analog substitution can also be performed for the non-affine key schedule, this then yields our `Rescue-XLIX` polynomial models.

Definition 2.4. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}^n_{\geq 1}$ be such that* $\gcd(d, q-1) = 1$ *and $r_{in}, r_{out} < n$, let $\mathbf{M}_0, \dots, \mathbf{M}_{2\cdot r} \in \mathbb{F}_q^n$ be invertible matrices, and let $\mathbf{c}_1, \dots, \mathbf{c}_{2\cdot r}, \mathbf{d}_1, \dots, \mathbf{d}_{2\cdot r} \in \mathbb{F}_q^n$ be constants. Let $\mathbf{x}_{in} = \left(x_{in,1}, \dots, x_{in,r_{in}}\right)^\intercal$, $\mathbf{x}_{out} = \left(x_{out,1}, \dots, x_{out,r_{out}}\right)^\intercal$, $\mathbf{x}^{(i)} = \left(x_1^{(i)}, \dots, x_n^{(i)}\right)^\intercal$, where $1 \leq i \leq 2\cdot r$, and $\mathbf{y}^{(j)} = \left(y_1^{(j)}, \dots, y_n^{(j)}\right)^\intercal$, where $0 \leq j \leq 2\cdot r$, be variables.*

(1) Let $\mathbf{p}, \mathbf{c} \in \mathbb{F}_q^n$ be a plain/ciphertext pair given by a `Rescue-XLIX` cipher function. In the polynomial ring $\mathbb{F}_q\left[\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2\cdot r)}, \mathbf{y}^{(0)}, \dots, \mathbf{y}^{(2\cdot r)}\right]$ let

$$\mathbf{f}_{cipher}^{(i)} = \begin{cases} \mathbf{M}_0 \left(\mathbf{p} + \mathbf{y}^{(0)}\right) - \mathbf{x}^{(1)}, & i = 0, \\ \mathbf{M}_i \mathbf{x}^{(i)} + \mathbf{y}^{(i)} + \mathbf{c}_i - \mathbf{x}^{(i+1)}, & \begin{Bmatrix} 1 \leq i \leq 2\cdot r - 1, \\ i \equiv 0 \mod 2 \end{Bmatrix}, \\ \mathbf{M}_{2\cdot r} \mathbf{x}^{(2\cdot r)} + \mathbf{y}^{(2\cdot r)} + \mathbf{c}_{2\cdot r} - \mathbf{c}, & i = 2\cdot r. \end{cases}$$

(a) For an affine key schedule, let $\hat{\mathbf{M}}_1, \dots, \hat{\mathbf{M}}_{2\cdot r} \in \mathbb{F}_q^{n\times n}$, and for $1 \leq i \leq 2\cdot r$ let

$$\mathbf{f}_{cipher}^{(i)} = \left\{ \mathbf{M}_i \mathcal{S}_d\left(\mathbf{x}^{(i)}\right) + \mathbf{y}^{(i)} + \mathbf{c}_i - \mathcal{S}_d\left(\mathbf{x}^{(i+1)}\right), \quad \begin{Bmatrix} 1 \leq i \leq 2\cdot r, \\ i \equiv 1 \mod 2 \end{Bmatrix} \right.,$$
$$\mathbf{k}^{(i)} = \hat{\mathbf{M}}_i \mathbf{y}^{(i-1)} + \mathbf{d}_i - \mathbf{y}^{(i)}.$$

(b) For a non-affine key schedule, let

$$\mathbf{f}_{cipher}^{(i)} = \left\{ \mathbf{M}_i \mathcal{S}_d\left(\mathbf{x}^{(i)}\right) + \mathcal{S}\left(\mathbf{y}^{(i)}\right) + \mathbf{c}_i - \mathcal{S}_d\left(\mathbf{x}^{(i+1)}\right), \quad \begin{Bmatrix} 1 \leq i \leq 2\cdot r, \\ i \equiv 1 \mod 2 \end{Bmatrix} \right.,$$
$$\mathbf{k}^{(i)} = \begin{cases} \mathbf{M}_i \mathcal{S}_d\left(\mathbf{y}^{(i-1)}\right) + \mathbf{d}_i - \mathcal{S}_d\left(\mathbf{y}^{(i)}\right), & \begin{Bmatrix} 1 \leq i \leq 2\cdot r, \\ i \equiv 1 \mod 2 \end{Bmatrix}, \\ \mathbf{M}_i \mathbf{y}^{(i-1)} + \mathbf{d}_i - \mathbf{y}^{(i)}, & \begin{Bmatrix} 1 \leq i \leq 2\cdot r, \\ i \equiv 0 \mod 2 \end{Bmatrix}. \end{cases}$$

The polynomial system $\mathcal{F}_{cipher} = \left\{\mathbf{f}_{cipher}^{(i)}\right\}_{0 \leq i \leq 2\cdot r} \cup \left\{\mathbf{k}^{(j)}\right\}_{1 \leq j \leq 2\cdot r}$ is called the `Rescue-XLIX` polynomial system.

Let $\boldsymbol{\alpha} \in \mathbb{F}_q^{n - r_{in}}$ and $\boldsymbol{\beta} \in \mathbb{F}_q^{n - r_{out}}$, in the polynomial ring $\mathbb{F}_q\left[\mathbf{x}_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2\cdot r)}, \mathbf{x}_{out}\right]$, let

$$\mathbf{f}_{sponge}^{(i)} = \begin{cases} \mathbf{M}_0 \begin{pmatrix} \mathbf{x}_{in} \\ \boldsymbol{\alpha} \end{pmatrix} - \mathbf{x}^{(1)}, & i = 0, \\ \mathbf{M}_i \mathcal{S}_d\left(\mathbf{x}^{(i)}\right) + \mathbf{c}_i - \mathcal{S}_d\left(\mathbf{x}^{(i+1)}\right), & \begin{Bmatrix} 1 \leq i \leq 2\cdot r - 1, \\ i \equiv 1 \mod 2, \end{Bmatrix} \\ \mathbf{M}_i \mathbf{x}^{(i)} + \mathbf{c}_i - \mathbf{x}^{(i+1)}, & \begin{Bmatrix} 1 \leq i \leq 2\cdot r - 1, \\ i \equiv 0 \mod 2. \end{Bmatrix} \end{cases}$$

(2) *Let*

$$\mathbf{f}_{pre}^{(2 \cdot r)} = \mathbf{M}_{2 \cdot r} \mathbf{x}^{(2 \cdot r)} + \mathbf{c}_{2 \cdot r} - \begin{pmatrix} \boldsymbol{\beta} \\ \mathbf{x}_{out} \end{pmatrix}.$$

The polynomial system $\mathcal{F}_{pre} = \left\{ \mathbf{f}_{sponge}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r - 1} \cup \left\{ \mathbf{f}_{pre}^{(2 \cdot r)} \right\}$ *is called the* ***`Rescue-XLIX-Prime`*** *preimage polynomial system.*

(3) *Let*

$$\mathbf{f}_{pre}^{(2 \cdot r)} = \mathbf{M}_{2 \cdot r} \mathbf{x}^{(2 \cdot r)} + \mathbf{c}_{2 \cdot r} - \begin{pmatrix} \mathbf{x}_{out} \\ \boldsymbol{\beta} \end{pmatrix}.$$

The polynomial system $\mathcal{F}_{CICO} = \left\{ \mathbf{f}_{sponge}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r - 1} \cup \left\{ \mathbf{f}_{CICO}^{(2 \cdot r)} \right\}$ *is called the* ***`Rescue-XLIX-Prime`*** *CICO polynomial system.*

Remark 2.5. Since `Rescue` starts with $x^{\frac{1}{d}}$, we cannot apply the substitution of `Rescue-XLIX` for the 1st and the 2nd round. We can only apply it to the 2nd and the 3rd round, the 4th and the 5th round, etc. In particular, for the fist round of `Rescue` we cannot omit the auxiliary equations, and the last round is non-linear since it represents the SPN for x^d. Therefore, the `Rescue` polynomial models have a slightly less efficient representation than the ones for `Rescue-XLIX`.

2.3 Term Orders & Gröbner Bases

Let $P = K[x_1, \ldots, x_n]$, and let $m = \prod_{i=1}^{n} x_i^{a_i} \in P$ be a monomial. The monomial m can be identified with the integer vector $\mathbf{a} = (a_1, \ldots, a_n)^\intercal \in \mathbb{Z}_{\geq 0}^n$. Since we can sort integer vectors, we can define term orders on P, i.e. a binary relation to sort the monomials in P.

Definition 2.6 (cf. [CLO15, Chapter 2 §2 Definition 1])**.** *Let K be a field, a term order $>$ on $K[x_1, \ldots, x_n]$ is a relation $>$ on $\mathbb{Z}_{\geq 0}^n$ such that*

(i) *$>$ is a total ordering on $\mathbb{Z}_{\geq 0}$.*

(ii) *If $\mathbf{a} > \mathbf{b}$ and $\mathbf{c} \in \mathbb{Z}_{\geq 0}^n$, then $\mathbf{a} + \mathbf{c} > \mathbf{b} + \mathbf{c}$.*

(iii) *$>$ is a well-ordering on $\mathbb{Z}_{\geq 0}^n$, i.e. every non-empty subset of $\mathbb{Z}_{\geq 0}^n$ has a smallest element under $>$.*

Let us recall the standard examples of term orders.

Example 2.7. Let $\mathbf{a} = (a_1, \ldots, a_n)^\intercal, \mathbf{b} = (b_1, \ldots, b_n)^\intercal \in \mathbb{Z}_{\geq 0}^n$.

(1) We say that lexicographically $\mathbf{a} >_{LEX} \mathbf{b}$ if the first non-zero entry of $\mathbf{a} - \mathbf{b}$ is positive. We denote this term order as LEX.

(2) We say that reverse lexicographically $\mathbf{a} >_{RLEX} \mathbf{b}$ if the last non-zero entry of $\mathbf{a} - \mathbf{b}$ is negative. We denote this term order as RLEX.

(3) We say that (degree) graded lexicographically $\mathbf{a} >_{DLEX} \mathbf{b}$ if $\sum_{i=1}^{n} a_i > \sum_{i=1}^{n} b_i$ or $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i$ and $\mathbf{a} >_{LEX} \mathbf{b}$.

(4) We say that (degree) graded reverse lexicographically $\mathbf{a} >_{DRL} \mathbf{b}$ if $\sum_{i=1}^{n} a_i > \sum_{i=1}^{n} b_i$ or $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i$ and $\mathbf{a} >_{RLEX} \mathbf{b}$. We denote this term order as DRL.

Moreover, we can associate weights to the variables to yield new term orders, so-called *weight orders*.

Definition 2.8. *Let $\mathbf{w} \in \mathbb{R}_{\geq 0}^n$, and let $>_\tau$ be a term order. For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_{\geq 0}^n$, the weight order $\mathbf{a} >_{\mathbf{w}, \tau} \mathbf{b}$ is defined as* *(i) If* $\langle \mathbf{w}, \mathbf{a} \rangle > \langle \mathbf{w}, \mathbf{b} \rangle$*, then* $\mathbf{a} >_{\mathbf{w},\tau} \mathbf{b}$*.*

(ii) If $\langle \mathbf{w}, \mathbf{a} \rangle = \langle \mathbf{w}, \mathbf{b} \rangle$*, then* $\mathbf{a} >_{\tau} \mathbf{b}$*.*

We call $\mathbf{w}$ the weight or weight vector, and $>_\tau$ the base order. The simplest example of a weight order it the graded lexicographic order which has $\mathbf{w} = (1, \dots, 1)^\intercal$.

The base order can be a weight order itself, then we can collect the weights in a matrix. Suppose that we have an iterated sequence of m weight orders, then we can collect the weights in the matrix $\mathbf{W} \in \mathbb{R}_{\geq 0}^{m \times n}$. We denote such a weight order as $>_{\mathbf{W},\tau}$, where τ is some base order and $\mathbf{W}$ is called the weight matrix. Given $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_{\geq 0}^n$ we one decides $\mathbf{a} >_{\mathbf{W},\tau} \mathbf{b}$ or not via the following iteration:

(1) Compute $\hat{\mathbf{a}} = \mathbf{W}\mathbf{a}$ and $\hat{\mathbf{b}} = \mathbf{W}\mathbf{b}$, and set $i = 1$.

(2) If $\hat{a}_i > \hat{b}_i$, then $\mathbf{a} >_{\mathbf{W},\tau} \mathbf{b}$.

(3) Else $i \mapsto i + 1$, if $i \leq m$ return to Step (2) else move to Step (4).

(4) Fall back to the base order $>_\tau$ to decide whether $\mathbf{a} >_\tau \mathbf{b}$ or not.

Simplest example of such a generalized weight order is the LEX order which has weight matrix $\mathbf{W} = \mathbf{I}_{n \times n}$. In particular, all the term orders from Example 2.7 can be represented via a weight matrix. It is worthwhile mentioning that every term order can be constructed via an iteration of weight orders [Rob86].

For ease of writing, we will work with variable vectors most of the time in this paper, see e.g. Definition 2.4. For a term order $>$, if we write $\mathbf{x} > \mathbf{y}$, then this shall be understood as $x_1 > \ldots > x_n > y_1 > \ldots > y_n$.

2.3.1 Gröbner Bases

Gröbner bases are a fundamental concept in computer algebra, they were introduced in Bruno Buchberger's PhD thesis [Buc65]. With Gröbner bases one can solve many computational problems for polynomial ideals, most interesting to us is the computation of the set of zeros of a zero-dimensional ideal. Let $f \in P = K[x_1, \dots, x_n]$ be a polynomial, and let $>$ be a term order on P, the leading monomial $\mathrm{LM}_>(f)$ is the largest monomial with non-zero coefficient in f with respect to $>$. For an ideal $I \subset P$, a $>$-Gröbner is a finite set of generators $\mathcal{G} \subset I$ such that

$$\big(\mathrm{LM}_>(f) \mid f \in I \big) = \big(\mathrm{LM}_>(g) \mid g \in \mathcal{G} \big). \tag{13}$$

For a general introduction into the theory of Gröbner bases we refer to [KR00, KR05, CLO15].

One can verify that a finite set of generators is a $>$-Gröbner via Buchberger's criterion [CLO15, Chapter 2 §6 Theorem 6]. Like in [Ste24b] we are only interested in a special case of Buchberger's criterion: Pairwise coprime leading monomials under $>$ implies being a $>$-Gröbner basis.

Lemma 2.9 ([Ste24b, Lemma 2.10])**.** *Let* K *be a field, let* $\mathcal{F} = \{f_1, \dots, f_m\} \subset P = K[x_1, \dots, x_n]$*, and let* $>$ *be a term order on* P*. If for all* $i \neq j$

$$\gcd \big(\mathrm{LM}_>(f_i), \mathrm{LM}_>(f_j) \big) = 1,$$

then $\mathcal{F}$ *is a* $>$*-Gröbner basis.*

Proof. This is an immediate consequence of [CLO15, Chapter 2 §9 Theorem 3, Proposition 4]. □

3 Rescue-XLIX Gröbner Bases

3.1 Cipher

As pointed out in [Ste24a, §6.4] non-affine key schedules break structure that is necessary to construct a DRL Gröbner basis for a SPN cipher. For Rescue-XLIX we fix this problem by constructing a dedicated weight order. Essentially, we define one weight vector which is supposed to decide for the key schedule polynomials or enforce ties, and $2 \cdot r$ weight vectors which decide for exactly one round of Rescue-XLIX. It turns out that we can choose the same weights for the affine as well as the non-affine key schedule.

Theorem 3.1. *Let $\mathbb{F}_q$ be a finite field, let $n, r, d \in \mathbb{Z}_{\geq 1}$, and let $\mathcal{F}_{cipher} = \left\{ \mathbf{f}_{cipher}^{(i)} \right\}_{0 \leq i \leq r} \cup \left\{ \mathbf{k}^{(j)} \right\}_{1 \leq j \leq r} \subset \mathbb{F}_q \left[\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, \mathbf{y}^{(0)}, \dots, \mathbf{y}^{(2 \cdot r)} \right]$ be parameters of a Rescue-XLIX cipher polynomial system. Let $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r} \in \mathbb{Z}^{n \cdot (4 \cdot r + 1)}$ be weight vectors defined as*

$$\mathbf{w}_0 = \begin{pmatrix} \mathbf{0}_{n \cdot 2 \cdot r} \\ \mathbf{1}_{n \cdot (2 \cdot r + 1)} \end{pmatrix}, \qquad \mathbf{w}_i = \begin{pmatrix} \mathbf{0}_{n \cdot (i-1)} \\ \mathbf{1}_n \\ \mathbf{0}_{n \cdot (2 \cdot r - i)} \\ \mathbf{0}_{n \cdot (2 \cdot r + 1)} \end{pmatrix},$$

where $1 \leq i \leq 2 \cdot r$, and let $\mathbf{W} = \begin{pmatrix} \mathbf{w}_0 & \dots & \mathbf{w}_{2 \cdot r} \end{pmatrix}^\intercal \in \mathbb{Z}^{(2 \cdot r + 1) \times n \cdot (4 \cdot r + 1)}$. Let $\mathbf{y}^{(2 \cdot r)} >_{LEX} > \ldots >_{LEX} \mathbf{y}^{(0)} >_{LEX} \mathbf{x}^{(1)} >_{LEX} > \ldots >_{LEX} \mathbf{x}^{(2 \cdot r)}$, and let $>_{\mathbf{W}, LEX}$ be a weight order on the Rescue-XLIX polynomial ring. Assume that for all $1 \leq i \leq 2 \cdot r$ the matrix $\mathbf{M}_i^{-1}$ has at least two non-zero entries in every row. Then

(1) A $>_{\mathbf{W}, LEX}$-Gröbner basis for $\mathcal{F}_{cipher}$ can be computed via linear transformations.

(2) For the quotient space dimension:

$$\dim_{\mathbb{F}_q} \left(\mathcal{F}_{cipher} \right) = \begin{cases} d^r, & \textit{affine key schedule}, \\ d^{2 \cdot r}, & \textit{non-affine key schedule}. \end{cases}$$

Proof. Let

$$\mathcal{G} = \left\{ \mathbf{g}^{(i)} = \mathbf{M}_i^{-1} \mathbf{f}_{\text{cipher}}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r} \cup \left\{ \mathbf{k}^{(j)} \right\}_{1 \leq j \leq 2 \cdot r},$$

we claim that $\mathcal{G}$ is a $>_{\mathbf{W}, LEX}$-Gröbner basis.

- For the weight $\mathbf{w}_0$, the variables $\mathbf{y}^{(i)}$ have weight 1 and the variables $\mathbf{x}^{(i)}$ have weight 0. Therefore,

  $$\mathrm{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{g}^{(0)} \right) = \mathbf{y}^{(0)}.$$

  For the $\mathbf{k}^{(j)}$'s:

  - For the affine key schedule, we have a trivial decision for a variable of $\mathbf{y}^{(j)}$ if $\hat{\mathbf{M}}_i$ has a zero row. Otherwise, we have a tie.
  - For the non-affine key schedule, we always have a tie, because the matrices $\mathbf{M}_i$ are invertible.

  Moreover, $\mathbf{w}_1, \dots, \mathbf{w}_{2 \cdot r}$ trivially produce ties on the $\mathbf{k}^{(j)}$'s, so we have to decide via LEX which yields for the affine key schedule that

  $$\mathrm{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{k}^{(j)} \right) = \mathbf{y}^{(j)},$$

  and for the non-affine key-schedule that

  $$\mathrm{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{k}^{(j)} \right) = \begin{cases} \mathcal{S}_d \left(\mathbf{y}^{(j)} \right), & j \equiv 1 \mod 2, \\ \mathbf{y}^{(j)}, & j \equiv 0 \mod 2. \end{cases}$$

- For the remaining $\mathbf{g}^{(i)}$'s, by assumption $\mathbf{M}_i^{-1}$ has at least two non-zero entries on every row. Therefore, at least two terms of $\mathbf{y}^{(i)}$ respectively $\mathcal{S}\left(\mathbf{y}^{(i)}\right)$ are present in every component of $\mathbf{g}^{(i)}$ and hence $\mathbf{w}_0$ produces ties. For $1 \leq j < i$, the variables $\mathbf{x}^{(i)}$, $\mathbf{x}^{(i+1)}$ (if $i < 2 \cdot r$) and $\mathbf{y}^{(i)}$ all have weight 0 with respect to $\mathbf{w}_j$, so we have ties. For $\mathbf{w}_i$, the variables $\mathbf{x}^{(i)}$ have weight 1 and the variables $\mathbf{x}^{(i+1)}$ and $\mathbf{y}^{(i)}$ have weight 0, therefore
$$\mathrm{LM}_{>_{\mathbf{w},LEX}}\left(\mathbf{g}^{(i)}\right) = \begin{cases} \mathcal{S}_d\left(\mathbf{x}^{(i)}\right), & i \equiv 1 \mod 2, \\ \mathbf{x}^{(i)}, & i \equiv 0 \mod 2. \end{cases}$$

Hence, all polynomials in $\mathcal{G}$ have pairwise coprime leading monomials, so by Lemma 2.9 we have constructed a Gröbner basis.

For a affine key schedule, the ideal of leading terms is given by
$$\left(\mathrm{LM}_{>_{\mathbf{w},LEX}}(f) \mid f \in \mathcal{G}\right) = \left(\mathbf{y}^{(0)}, \mathbf{y}^{(2\cdot j-1)}, \mathbf{y}^{(2\cdot j)}, \mathcal{S}_d\left(\mathbf{x}^{(2\cdot j-1)}\right), \mathbf{x}^{(2\cdot j)} \mid 1 \leq j \leq r\right),$$
and for the non-affine key schedule is given by
$$\left(\mathrm{LM}_{>_{\mathbf{w},LEX}}(f) \mid f \in \mathcal{G}\right) = \left(\mathbf{y}^{(0)}, \mathcal{S}_d\left(\mathbf{y}^{(2\cdot j-1)}\right), \mathbf{y}^{(2\cdot j)}, \mathcal{S}_d\left(\mathbf{x}^{(2\cdot j-1)}\right), \mathbf{x}^{(2\cdot j)} \mid 1 \leq j \leq r\right).$$
Counting the monomials not contained in these ideals yields the claim. □

3.2 Sponge

Recall that for SPN sponge functions [Ste24b, §3.1] one has to construct weights $\mathbf{w}_0, \dots, \mathbf{w}_r$ such that in the i^{th} round $\mathbf{w}_i$ separates the terms $\mathcal{S}\left(\mathbf{x}^{(i)}\right)$ and $\mathbf{x}^{(i+1)}$ into an input part of size r_{in} and an output part of size r_{out}. Moreover, $\mathbf{w}_i$ has to produce ties on all other rounds. To achieve the latter for a SPN, the weights in $\mathbf{w}_i$ increase iteratively for every round. In principle, we are going to apply this strategy to `Rescue-XLIX-Prime`, but we do not need to iteratively increase the weights, since in one round the non-linear terms all have the same degree.

Also, recall that for the construction of SPN sponge Gröbner bases the matrices had to exhibit certain "non-singularity" positions. For `Rescue-XLIX-Prime` we can use the same conditions to construct the Gröbner bases. For their formalization we require a linear map.

Definition 3.2 ([Ste24b, Definition 3.1])**.** *Let K be a field, let $k, l, m, n \in \mathbb{Z}_{\geq 1}$ be integers such that $k \leq m$ and $l \leq n$, and let*
$$\begin{aligned} \rho_{k,l} : K^{m \times n} &\to K^{k \times l}, \\ \mathbf{M} &\mapsto \begin{pmatrix} \mathbf{I}_{k \times l} & \mathbf{0}_{k \times (n-l)} \\ \mathbf{0}_{(m-k) \times l} & \mathbf{0}_{(m-k) \times (n-l)} \end{pmatrix} \mathbf{M}. \end{aligned}$$

Let us now recall the non-singular $\rho_{k,k}$-positions.

Definition 3.3 ([Ste24b, Definition 3.3])**.** *Let K be a field, let $k, n \in \mathbb{Z}_{\geq 1}$ be such that $k < n$, and let $\mathbf{M} \in K^{n \times n}$ be a matrix such that* $\operatorname{rank}\left(\rho_{k,k}(\mathbf{M})\right) = k$*. Then there exists an invertible matrix $\mathbf{N} \in K^{k \times k}$ such that*
$$\begin{pmatrix} \mathbf{N} & \mathbf{0}_{k \times (n-k)} \\ \mathbf{0}_{(n-k) \times k} & \mathbf{I}_{(n-k) \times (n-k)} \end{pmatrix} \mathbf{M} = \begin{pmatrix} \mathbf{I}_{k \times k} & \mathbf{A} \\ \mathbf{B} & \mathbf{C} \end{pmatrix},$$
where $\mathbf{A} \in K^{k \times (n-k)}$, $\mathbf{B} \in K^{(n-k) \times k}$ and $\mathbf{C} \in K^{(n-k) \times (n-k)}$.

(1) The matrix $\begin{pmatrix} \mathbf{N} & \mathbf{0}_{k \times (n-k)} \\ \mathbf{0}_{(n-k) \times k} & \mathbf{I}_{(n-k) \times (n-k)} \end{pmatrix}$ is called the $\rho_{k,k}$-transformation of $\mathbf{M}$.

(2) The matrix $\mathbf{M}$ is said to be in upper non-singular $\rho_{k,k}$-position if every row of $\mathbf{A}$ is non-zero.

(3) The matrix $\mathbf{M}$ is said to be in strong upper non-singular $\rho_{k,k}$-position if every row of $\mathbf{A}$ has at least two non-zero entries.

(4) The matrix $\mathbf{M}$ is said to be in lower non-singular $\rho_{k,k}$-position if every row of $\mathbf{B}$ is non-zero.

(5) The matrix $\mathbf{M}$ is said to be in strong lower non-singular $\rho_{k,k}$-position if every row of $\mathbf{B}$ has at least two non-zero entries.

3.2.1 Preimage

Now we construct a Gröbner basis for the `Rescue-XLIX-Prime` preimage polynomial system if $n > 2$ and $r_{in} < n - 1$ in analogy to [Ste24b, Theorem 3.4].

Theorem 3.4. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}^n$ be such that $n = r_{in} + r_{out}$, and let $\mathcal{F}_{pre} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq r} \subset \mathbb{F}_q \left[\mathbf{x}_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, \mathbf{x}_{out} \right]$ be a `Rescue-XLIX-Prime` preimage polynomial system with parameters d, n, r, r_{in} and r_{out}. Let $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r} \in \mathbb{Z}^{n \cdot (2 \cdot r + 1)}$ be weight vectors defined as*

$$\mathbf{w}_0 = \begin{pmatrix} \mathbf{1}_{r_{in}} \\ \mathbf{0}_{r_{in}} \\ \mathbf{1}_{r_{out}} \\ \mathbf{1}_{n \cdot (2 \cdot r - 1)} \\ \mathbf{1}_{r_{out}} \end{pmatrix}, \qquad \mathbf{w}_i = \begin{pmatrix} \mathbf{0}_{r_{in}} \\ \mathbf{0}_{n \cdot (i-1)} \\ \mathbf{1}_{r_{in}} \\ \mathbf{0}_{r_{out}} \\ \mathbf{0}_{r_{in}} \\ \mathbf{1}_{r_{out}} \\ \mathbf{1}_{n \cdot (2 \cdot r - i - 1)} \\ \mathbf{1}_{r_{out}} \end{pmatrix}, \qquad \mathbf{w}_{2 \cdot r} = \begin{pmatrix} \mathbf{0}_{r_{in}} \\ \mathbf{0}_{n \cdot (2 \cdot r - 1)} \\ \mathbf{1}_{r_{in}} \\ \mathbf{0}_{r_{out}} \\ \mathbf{1}_{r_{out}} \end{pmatrix},$$

where $1 \leq i \leq 2 \cdot r - 1$, and let $\mathbf{W} = \begin{pmatrix} \mathbf{w}_0 & \dots & \mathbf{w}_{2 \cdot r} \end{pmatrix}^\intercal \in \mathbb{Z}^{(2 \cdot r + 1) \times n \cdot (2 \cdot r + 1)}$. Let $\mathbf{x}_{out} >_{LEX} \mathbf{x}^{(2 \cdot r)} >_{LEX} \dots >_{LEX} \mathbf{x}^{(1)} > \mathbf{x}_{in}$, and let $>_{\mathbf{W},LEX}$ be a weight order on the `Rescue-XLIX-Prime` polynomial ring. Assume that

(i) $n > 2$,

(ii) $r_{in} < n - 1$,

(iii) $\operatorname{rank} \left(\rho_{r_{in}, r_{in}} (\mathbf{M}_0) \right) = r_{in}$,

(iv) $\mathbf{M}_i$ *is in upper non-singular $\rho_{r_{in}, r_{in}}$-position for all $1 \leq i \leq 2 \cdot r - 1$, and*

(v) $\mathbf{M}_{2 \cdot r}$ *is in strong upper non-singular $\rho_{r_{in}, r_{in}}$-position.*

Then

(1) A $>_{\mathbf{W},LEX}$-Gröbner basis for $\mathcal{F}_{pre}$ can be computed via linear transformations.

(2) $\dim_{\mathbb{F}_q} \left(\mathcal{F}_{pre} \right) = d^{n \cdot r}$.

Proof. By assumption $\operatorname{rank} \left(\rho_{r_{in}, r_{in}} (\mathbf{M}_i) \right) = r_{in}$ for all $0 \leq i \leq 2 \cdot r$, so we can find invertible matrices $\mathbf{N}_i \in \mathbb{F}_q^{r_{in} \times r_{in}}$ such that

$$\begin{pmatrix} \mathbf{N}_i & \mathbf{0}_{r_{in} \times r_{out}} \\ \mathbf{0}_{r_{out} \times r_{in}} & \mathbf{I}_{r_{out} \times r_{out}} \end{pmatrix} \mathbf{M}_i = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_i \\ \mathbf{B}_i & \mathbf{C}_i \end{pmatrix},$$

where $\mathbf{A}_i \in \mathbb{F}_q^{r_{in} \times r_{out}}$, $\mathbf{B}_i \in \mathbb{F}_q^{r_{out} \times r_{in}}$ and $\mathbf{C}_i \in \mathbb{F}_q^{r_{out} \times r_{out}}$. In addition, for all $1 \leq i \leq 2 \cdot r - 1$ the matrix $\mathbf{A}_i$ has all rows non-zero, and for $i = 2 \cdot r$ the matrix $\mathbf{A}_{2 \cdot r}$ has at least two non-zero entries on every row.

Now let

$$\mathcal{G} = \left\{ \mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{N}_i & \mathbf{0}_{r_{in} \times r_{out}} \\ \mathbf{0}_{r_{out} \times r_{in}} & \mathbf{I}_{r_{out} \times r_{out}} \end{pmatrix} \mathbf{f}_{\text{pre}}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r},$$

we claim that $\mathcal{G}$ is the $>_{\mathbf{w},LEX}$-Gröbner basis of $\mathcal{F}_{\text{pre}}$.

- For $i = 0$, we have that

  $$\mathbf{g}^{(0)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_0 \\ \mathbf{B}_0 & \mathbf{C}_0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_{in} \\ \boldsymbol{\alpha} \end{pmatrix} - \mathbf{x}^{(1)},$$

  For $\mathbf{w}_0$, the terms $\mathbf{x}_{in}$ and $\mathbf{x}^{(1)}|_{r_{out}}$ have weight 1 and the terms $\mathbf{x}^{(1)}|^{r_{in}}$ have weight 0. Therefore,

  $$\mathrm{LM}_{>_{\mathbf{w},LEX}} \left(\mathbf{g}^{(0)} \Big|^{r_{in}} \right) = \mathbf{x}_{in}$$

  On the other hand, for $\mathbf{g}^{(0)}|_{r_{out}}$ if $\mathbf{B}_0$ has a zero row, then we have a decision for a term of $\mathbf{x}^{(1)}|_{r_{out}}$, else we have ties.

  For $\mathbf{w}_1, \dots, \mathbf{w}_{2 \cdot r}$ all terms in $\mathbf{g}^{(0)}|_{r_{out}}$ have weight 0, so we have trivial ties, and we finally have to decide via LEX which yields

  $$\mathrm{LM}_{>_{\mathbf{w},LEX}} \left(\mathbf{g}^{(0)} \Big|_{r_{out}} \right) = \mathbf{x}^{(1)} \Big|_{r_{out}}.$$

- For $1 \leq i < 2 \cdot r$, let $0 \leq j < i - 1$.

  - If $i \equiv 1 \mod 2$, then

    $$\mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_i \\ \mathbf{B}_i & \mathbf{C}_i \end{pmatrix} \mathcal{S}_d \left(\mathbf{x}^{(i)} \right) + \hat{\mathbf{c}}_i - \begin{pmatrix} \mathbf{N}_i \mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) \Big|^{r_{in}} \\ \mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) \Big|_{r_{out}} \end{pmatrix},$$

    where $\hat{\mathbf{c}}_i \in \mathbb{F}_q^n$. For $\mathbf{w}_j$, the terms $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right)$ and $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right)$ both have weight d. On $\mathbf{g}^{(i)}|^{r_{in}}$, one term of $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right)$ and one of $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right)$ is always present, so we have ties. On $\mathbf{g}^{(i)}|_{r_{out}}$, if $\begin{pmatrix} \mathbf{B}_i & \mathbf{C}_i \end{pmatrix}$ has a zero row, then we have a trivial decision for a term of $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) |_{r_{out}}$, else we have ties.

    For $\mathbf{w}_{i-1}$, the terms $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right) |^{r_{in}}$ have weight 0, and the terms $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right) |_{r_{out}}$ and $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right)$ have weight d. Since $\mathbf{A}_i$ has all rows non-zero, at least one term of $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right) |_{r_{out}}$ and $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right)$ is always present in every component of $\mathbf{g}^{(i)}|^{r_{in}}$, so we have ties. On $\mathbf{g}^{(i)}|_{r_{out}}$, if $\mathbf{C}_i$ has a zero row, then we have a trivial decision for a term of $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) |_{r_{out}}$, else we have ties.

    Finally, decision via $\mathbf{w}_i$ yields that

    $$\mathrm{LM}_{>_{\mathbf{w},LEX}} \left(\mathbf{g}^{(i)} \Big|^{r_{in}} \right) = \mathcal{S}_d \left(\mathbf{x}^{(i)} \right) \Big|^{r_{in}}.$$

    For $\mathbf{g}^{(i)}|_{r_{out}}$, depending on whether $\mathbf{B}_i$ has a zero row or not we either have a trivial decision for a term of $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) |_{r_{out}}$ on that row, or we have again a tie.

    For $\mathbf{w}_{i+1}, \dots, \mathbf{w}_{2 \cdot r}$, the terms $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right)$ and $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) |_{r_{out}}$ have weight 0, so we trivially have a tie, and we finally have to decide via LEX which yields

    $$\mathrm{LM}_{>_{\mathbf{w},LEX}} \left(\mathbf{g}^{(i)} \Big|_{r_{out}} \right) = \mathcal{S}_d \left(\mathbf{x}^{7(i+1)} \right) \Big|_{r_{out}}.$$

  - If $i \equiv 0 \mod 2$, then
$$\mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_i \\ \mathbf{B}_i & \mathbf{C}_i \end{pmatrix} \mathbf{x}^{(i)} + \hat{\mathbf{c}}_i - \begin{pmatrix} \mathbf{N}_i \mathbf{x}^{(i+1)} \big|^{r_{in}} \\ \mathbf{x}^{(i+1)} \big|_{r_{out}} \end{pmatrix},$$
    where $\hat{\mathbf{c}}_i \in \mathbb{F}_q^n$. The arguments are identical to the previous case, we just have to substitute $\mathcal{S}_d$ by $\mathcal{S}_1$ in the previous arguments.

- For $i = 2 \cdot r$, recall that
$$\mathbf{g}^{(2 \cdot r)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_{2 \cdot r} \\ \mathbf{B}_{2 \cdot r} & \mathbf{C}_{2 \cdot r} \end{pmatrix} \mathbf{x}^{(i)} + \hat{\mathbf{c}}_i - \begin{pmatrix} \mathbf{N}_{2 \cdot r} \boldsymbol{\beta} \\ \mathbf{x}_{r_{out}} |_{r_{out}} \end{pmatrix},$$
  where $\hat{\mathbf{c}}_{2 \cdot r} \in \mathbb{F}_q^n$. Since $\mathbf{A}_{2 \cdot r}$ has at least two non-zero entries on every row, the weights $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r - 2}$ produce ties analog to the previous cases on $\mathbf{g}^{(2 \cdot r)} |^{r_{in}}$. For $\mathbf{g}^{(2 \cdot r)} |_{r_{out}}$, if $\begin{pmatrix} \mathbf{B}_{2 \cdot r} & \mathbf{C}_{2 \cdot r} \end{pmatrix}$ has a zero row, then we have a trivial decision for a term of $\mathbf{x}_{out}$, else we have ties.

  For $\mathbf{w}_{2 \cdot r - 1}$, the terms $\mathbf{x}^{(2 \cdot r)} |^{r_{in}}$ have weight 0 and the terms $\mathbf{x}^{(2 \cdot r)} |_{r_{out}}$ and $\mathbf{x}_{r_{out}}$ have weight 1. Since every row of $\mathbf{A}_{2 \cdot r}$ has at least two non-zero entries every component of $\mathbf{g}^{(2 \cdot r)} |^{r_{in}}$ has at least two terms of weight 1, so again we have a tie. For $\mathbf{g}^{(2 \cdot r)} |_{r_{out}}$, if $\mathbf{C}_{2 \cdot r}$ has a zero row, then we have a trivial decision for a term of $\mathbf{x}_{out}$ else we have ties.

  Finally, for $\mathbf{w}_{2 \cdot r}$ we have that
$$\mathrm{LM}_{>_{\mathbf{w}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \big|^{r_{in}} \right) = \mathbf{x}^{(2 \cdot r)} \big|^{r_{in}}.$$
  For $\mathbf{g}^{(2 \cdot r)} |_{r_{out}}$, if $\mathbf{B}_{2 \cdot r}$ has a zero row, then we have a trivial decision for a term of $\mathbf{x}_{out}$, else we have ties. In case of a tie, we have to do the final decision via LEX which yields
$$\mathrm{LM}_{>_{\mathbf{w}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \big|_{r_{out}} \right) = \mathbf{x}_{out}.$$

So all polynomials of $\mathcal{G}$ have pairwise coprime leading monomials which implies being a Gröbner basis by Lemma 2.9.

Let us compute the ideal of leading terms
$$\begin{aligned} &\big(\mathrm{LM}_{>_{\mathbf{w}, LEX}} (f) \mid f \in \mathcal{G} \big) \\ &= \left(\mathbf{x}_{in}, \mathcal{S}_d \left(\mathbf{x}^{(2 \cdot j - 1)} \right) \Big|^{r_{in}}, \mathbf{x}^{(2 \cdot j - 1)} \Big|_{r_{out}}, \mathbf{x}^{(2 \cdot j)} |^{r_{in}}, \mathcal{S}_d \left(\mathbf{x}^{(2 \cdot j)} \right) \Big|_{r_{out}}, \mathbf{x}_{out} \;\Bigg|\; 1 \leq j \leq r \right), \end{aligned}$$
and the claim for the quotient space dimension follows. □

Note that this proof fails if $n > 2$ and $r_{in} = n - 1$ for the last round, since $\mathbf{A}_r$ is a column vector then, i.e. it has only one entry on every row. Analog to [Ste24b, Proposition 3.5] we can fix this by modifying the weights.

Proposition 3.5. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}$ be integers such that $n = r_{in} + r_{out}$, and let $\mathcal{F}_{pre} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r} \subset \mathbb{F}_q \big[\mathbf{x}_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, \mathbf{x}_{out} \big]$ be a `Rescue-XLIX-Prime` preimage polynomial system with the parameters d, n, r, r_{in} and r_{out}. Let $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r} \in \mathbb{Z}_{\geq 0}^{n \cdot (2 \cdot r + 1)}$ be weight vectors defined as*
$$\mathbf{w}_0 = \begin{pmatrix} \mathbf{1}_{r_{in}} \\ \mathbf{0}_n \\ \mathbf{1}_{n \cdot (2 \cdot r - 1)} \\ \mathbf{1}_{r_{out}} \end{pmatrix}, \qquad \mathbf{w}_i = \begin{pmatrix} \mathbf{0}_{r_{in}} \\ \mathbf{0}_{n \cdot (i - 1)} \\ \mathbf{1}_{r_{in}} \\ \mathbf{0}_{r_{out}} \\ \mathbf{0}_n \\ \mathbf{1}_{n \cdot (2 \cdot r - i - 1)} \\ \mathbf{1}_{r_{out}} \end{pmatrix}, \qquad \mathbf{w}_{2 \cdot r} = \begin{pmatrix} \mathbf{0}_{r_{in}} \\ \mathbf{0}_{n \cdot (2 \cdot r - 1)} \\ \mathbf{1}_{r_{in}} \\ \mathbf{0}_{r_{out}} \\ \mathbf{0}_{r_{out}} \end{pmatrix},$$

where $1 \leq i \leq 2 \cdot r - 1$, and let $\mathbf{W} = \begin{pmatrix} \mathbf{w}_0 & \dots & \mathbf{w}_{2 \cdot r} \end{pmatrix}^\intercal \in \mathbb{Z}_{\geq 0}^{(2 \cdot r + 1) \times n \cdot (2 \cdot r + 1)}$. Let $x_{out} >_{LEX} \mathbf{x}^{(2 \cdot r)} >_{LEX} \ldots >_{LEX} \mathbf{x}^{(1)} >_{LEX} \mathbf{x}_{in}$, and let $>_{\mathbf{W},LEX}$ be a weight order on the ***`Rescue-XLIX-Prime`*** *polynomial ring. Assume that*

(i) $n > 2$,

(ii) $r_{in} = n - 1$,

(iii) the matrix $\mathbf{M}_0$ is in strong lower non-singular $\rho_{r_{in}, r_{in}}$-position,

(iv) for all $1 \leq i \leq 2 \cdot r - 1$:

   (a) $\operatorname{rank}\left(\rho_{r_{in}, r_{in}} (\mathbf{M}_i) \right) = r_{in}$,

   (b) let $\mathbf{N}_i \in \mathbb{F}_q^{r_{in} \times r_{in}}$ be the matrix of the $\rho_{r_{in}, r_{in}}$-transformation of $\mathbf{M}_i$, then $\mathbf{N}_i$ has at least two non-zero entries in every row, and

(v) $\operatorname{rank}\left(\rho_{r_{in}, r_{in}} (\mathbf{M}_{2 \cdot r}) \right) = r_{in}$.

Then

(1) A $>_{\mathbf{W},LEX}$-Gröbner basis for $\mathcal{F}_{pre}$ can be computed via linear transformations.

(2) $\dim_{\mathbb{F}_q} \left(\mathcal{F}_{pre} \right) = d^{n \cdot r}$.

Proof. We consider the same $\mathcal{G}$ as in the proof of Theorem 3.4, we claim that it is the $>_{\mathbf{W},LEX}$-Gröbner basis.

- For $i = 0$, by choice of $\mathbf{w}_0$ the variables $\mathbf{x}_{in}$ have weight 1 and the ones of $\mathbf{x}^{(1)}|_{r_{out}}$ have weight 0. So we have that
$$\mathrm{LM}_{>_{\mathbf{W},LEX}} \left(\mathbf{g}^{(0)} \big|^{r_{in}} \right) = \mathbf{x}_{in}.$$
Moreover, since $\mathbf{M}_0$ is in strong lower non-singular $\rho_{r_{in}, r_{in}}$-position, every row of $\mathbf{B}_0$ has at least two non-zero entries, therefore at least two terms of $\mathbf{x}_{in}$ are present in every component of $\mathbf{g}^{(0)}|_{r_{out}}$, so we have a tie.
But for $\mathbf{w}_1, \ldots, \mathbf{w}_{2 \cdot r}$ we have trivial ties, so we have to decide via LEX which yields
$$\mathrm{LM}_{>_{\mathbf{W},LEX}} \left(\mathbf{g}^{(0)} \big|^{r_{out}} \right) = \mathbf{x}^{(1)} \Big|_{r_{out}}.$$
- For $1 \leq i < 2 \cdot r$, the weights $\mathbf{w}_0, \ldots, \mathbf{w}_{i-2}$ produce a tie analog to Theorem 3.4. For $\mathbf{w}_{i-1}$:
  - If $i \equiv 1 \mod 2$, recall that
$$\mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_i \\ \mathbf{B}_i & \mathbf{C}_i \end{pmatrix} \mathcal{S}_d \left(\mathbf{x}^{(i)} \right) + \hat{\mathbf{c}}_i - \begin{pmatrix} \mathbf{N}_i \mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) \Big|^{r_{in}} \\ \mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) \Big|_{r_{out}} \end{pmatrix},$$
where $\hat{\mathbf{c}}_i \in \mathbb{F}_q^n$. The terms $\mathcal{S}_d \left(\mathbf{x}^{(i)} \right)$ have weight 0 and the terms $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right)$ have weight d with respect to $\mathbf{w}_{i-1}$. Therefore,
$$\mathrm{LM}_{>_{\mathbf{W},LEX}} \left(\mathbf{g}^{(i)} \big|_{r_{out}} \right) = \mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) \Big|_{r_{out}}.$$
On the other hand, by assumption $\mathbf{N}_i$ has at least two non-zero entries on every row, so at least two terms of $\mathcal{S}_d \left(\mathbf{x}^{(i+1)} \right) |^{r_{in}}$ are present in every component of $\mathbf{g}^{(i)}|^{r_{in}}$, so we have a tie.
Finally, for $\mathbf{w}_i$ we trivially have that
$$\mathrm{LM}_{>_{\mathbf{W},LEX}} \left(\mathbf{g}^{(i)} \big|^{r_{in}} \right) = \mathcal{S}_d \left(\mathbf{x}^{(i)} \right) \Big|^{r_{in}}.$$

  - If $i \equiv 0 \mod 2$, recall that

$$\mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_i \\ \mathbf{B}_i & \mathbf{C}_i \end{pmatrix} \mathbf{x}^{(i)} + \hat{\mathbf{c}}_i - \begin{pmatrix} \mathbf{N}_i \mathbf{x}^{(i+1)} \big|^{r_{in}} \\ \mathbf{x}^{(i+1)} \big|_{r_{out}} \end{pmatrix},$$

    where $\hat{\mathbf{c}}_i \in \mathbb{F}_q^n$. The arguments are identical to the previous case, we just have to substitute $\mathcal{S}_d$ by $\mathcal{S}_1$ in the previous arguments.

- For $i = 2 \cdot r$, recall that

$$\mathbf{g}^{(2 \cdot r)} = \begin{pmatrix} \mathbf{I}_{r_{in} \times r_{in}} & \mathbf{A}_{2 \cdot r} \\ \mathbf{B}_{2 \cdot r} & \mathbf{C}_{2 \cdot r} \end{pmatrix} \mathbf{x}^{(2 \cdot r)} + \hat{\mathbf{c}}_{2 \cdot r} - \begin{pmatrix} \mathbf{N}_{2 \cdot r} \boldsymbol{\beta} \\ \mathbf{x}_{r_{out}} |_{r_{out}} \end{pmatrix},$$

  where $\hat{\mathbf{c}}_{2 \cdot r} \in \mathbb{F}_q^n$. Depending on whether $\mathbf{A}_{2 \cdot r}$ has a zero row or not, we either have a trivial decision for a term of $\mathbf{x}^{(2 \cdot r)}|^{r_{in}}$, or the weights $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r - 2}$ produce ties analog to the previous case.

  For $\mathbf{w}_{r-1}$, the terms $\mathbf{x}^{(2 \cdot r)}$ have weight 0 and the ones of $\mathbf{x}_{out}$ have weight 1, so we have that

$$\mathrm{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \big|_{r_{out}} \right) = x_{out}.$$

  But on $\mathbf{g}^{(2 \cdot r)}|^{r_{in}}$ all terms have weight 0, so we have to decide via $\mathbf{w}_{2 \cdot r}$ which yields

$$\mathrm{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \big|^{r_{in}} \right) = \mathbf{x}^{(2 \cdot r)} \big|^{r_{in}}.$$

By Lemma 2.9, pairwise coprime leading monomials of $\mathcal{G}$ implies being a Gröbner basis.

Counting the number of monomials not contained in the ideal of leading terms is analog to Theorem 3.4. □

The previous proofs also fail for $n = 2$, since N_i, A_i, B_i and C_i are single field elements, so none of the previous conditions can be satisfied. Luckily, we can again fix this analog to [Ste24b, Corollary 3.6] by considering the weights from Theorem 3.4 and modifying the LEX order as well as the polynomials in the last round a bit.

Corollary 3.6. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}$ be integers such that $n = 2$, $r_{in} = 1$ and $n = r_{in} + r_{out}$, and let $\mathcal{F}_{pre} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r} \subset \mathbb{F}_q \left[x_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, x_{out} \right]$ be a* **`Rescue-XLIX-Prime`** *preimage polynomial system with the parameters d, n, r, r_{in} and r_{out}. Let $\mathbf{W} \in \mathbb{Z}_{\geq 0}^{(2 \cdot r + 1) \times n \cdot (2 \cdot r + 1)}$ be the weight matrix from Theorem 3.4, let $x_1^{(2 \cdot r)} >_{LEX} x_{out} >_{LEX} x_2^{(2 \cdot r)} >_{LEX} \mathbf{x}^{(2 \cdot r - 1)} >_{LEX} \dots >_{LEX} \mathbf{x}^{(1)} >_{LEX} x_{in}$, and let $>_{\mathbf{W}, LEX}$ be a weight order on the* **`Rescue-XLIX-Prime`** *polynomial ring. Assume that $\mathbf{M}_i$ is in upper non-singular $\rho_{r_{in}, r_{in}}$-position for all $0 \leq i \leq r$. Then*

(1) A $>_{\mathbf{W}, LEX}$-Gröbner basis for $\mathcal{F}_{pre}$ can be computed via linear transformations.

(2) $\dim_{\mathbb{F}_q} \left(\mathcal{F}_{pre} \right) = d^{2 \cdot r}$.

Proof. Let $\mathbf{g}^{(0)}, \dots, \mathbf{g}^{(2 \cdot r)}$ be as in Theorem 3.4, we introduce a minor modification in the last round

$$\begin{aligned} \hat{\mathbf{g}}^{(2 \cdot r)} &= \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -B_{2 \cdot r} & 1 \end{pmatrix} \begin{pmatrix} N_{2 \cdot r} & 0 \\ 0 & 1 \end{pmatrix} \mathbf{g}^{(2 \cdot r)} \\ &= \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -B_{2 \cdot r} & 1 \end{pmatrix} \begin{pmatrix} N_{2 \cdot r} & 0 \\ 0 & 1 \end{pmatrix} \left(\mathbf{M}_{2 \cdot r} \mathbf{x}^{(2 \cdot r)} + \mathbf{c}_{2 \cdot r} - \begin{pmatrix} \beta \\ x_{out} \end{pmatrix} \right) \\ &= \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & A_{2 \cdot r} \\ 0 & C_{2 \cdot r} - B_{2 \cdot r} \cdot A_{2 \cdot r} \end{pmatrix} \mathbf{x}^{(2 \cdot r)} + \hat{\mathbf{c}}_{2 \cdot r} - \begin{pmatrix} \beta \cdot N_{2 \cdot r} \\ x_{out} - B_{2 \cdot r} \cdot N_{2 \cdot r} \boldsymbol{\beta} \end{pmatrix} \right), \end{aligned}$$

where $\gamma \in \mathbb{F}_q^\times$ is chosen such that the coefficient of $x_2^{(2 \cdot r)}$ in $\hat{g}_1^{(2 \cdot r)}$ is non-zero. Then, $x_1^{(2 \cdot r)}$, $x_2^{(2 \cdot r)}$ and x_{out} are present in $g_1^{(2 \cdot r)}$, but only $x_2^{(2 \cdot r)}$ and x_{out} are present in $g_2^{(2 \cdot r)}$. We claim that

$$\mathcal{G} = \left\{ \mathbf{g}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r - 1} \cup \left\{ \hat{\mathbf{g}}^{(2 \cdot r)} \right\}$$

is the $>_{\mathbf{W}, LEX}$-Gröbner basis of $\mathcal{F}_{\text{pre}}$.

- For $0 \leq i \leq 2 \cdot r - 1$, the argument is identical to Theorem 3.4.
- For $i = r$, the term orders $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r - 2}$ produce ties for $\hat{\mathbf{g}}^{(2 \cdot r)}$ analog to Theorem 3.4.

  For $\mathbf{w}_{r-1}$, $x_1^{(2 \cdot r)}$ has weight 0, but $x_2^{(2 \cdot r)}$ and x_{out} have weight 1. By the construction of $\hat{\mathbf{g}}^{(2 \cdot r)}$, $x_2^{(2 \cdot r)}$ and x_{out} are present in both components, so we again produced ties.

  For $\mathbf{w}_{2 \cdot r}$, $x_1^{(2 \cdot r)}$ has weight 1, $x_2^{(2 \cdot r)}$ has weight 0 and x_{out} has weight 1. So trivially, we have that
  $$\text{LM}_{>_{\mathbf{W}, LEX}} \left(g_2^{(2 \cdot r)} \right) = x_{out}.$$
  For the first component we again have a tie, so we have to make the final decision via LEX which yields
  $$\text{LM}_{>_{\mathbf{W}, LEX}} \left(g_1^{(2 \cdot r)} \right) = x_1^{(2 \cdot r)}.$$

Again, by Lemma 2.9 we have constructed a Gröbner basis.

Counting the number of monomials not contained in the ideal of leading terms is analog to Theorem 3.4. □

3.2.2 CICO

The CICO polynomial system only differs in the last round from the preimage one. If we invert the matrix in the last round, and apply another transformation analog to Definition 3.3, then we reproduce the shape of preimage polynomial systems. To formalize the transformation we need to introduce a new linear map.

Definition 3.7 ([Ste24b, Definition 3.7])**.** *Let K be a field, let $k, l, m, n \in \mathbb{Z}_{\geq 1}$ be integers such that $k \leq m$ and $l \leq n$, and let*

$$\sigma_{k,l} : K^{m \times n} \to K^{k \times l},$$
$$\mathbf{M} \mapsto \begin{pmatrix} \mathbf{0}_{(m-k) \times (n-l)} & \mathbf{0}_{(m-k) \times l} \\ \mathbf{0}_{k \times (n-l)} & \mathbf{I}_{k \times l} \end{pmatrix} \mathbf{M}.$$

If in addition $m = n$ and $k = l$ and $\operatorname{rank}\left(\sigma_{k,k}(\mathbf{M})\right) = k$, then there exists an invertible matrix $\mathbf{N} \in K^{k \times k}$ such that

$$\begin{pmatrix} \mathbf{I}_{(n-k) \times (n-k)} & \mathbf{0}_{(n-k) \times k} \\ \mathbf{0}_{k \times (n-k)} & \mathbf{N} \end{pmatrix} \mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{I}_{k \times k} \end{pmatrix},$$

where $\mathbf{A} \in K^{(n-k) \times (n-k)}$, $\mathbf{B} \in K^{k \times (n-k)}$ and $\mathbf{C} \in K^{k \times (n-k)}$.
The matrix $\begin{pmatrix} \mathbf{I}_{(n-k) \times (n-k)} & \mathbf{0}_{(n-k) \times k} \\ \mathbf{0}_{k \times (n-k)} & \mathbf{N} \end{pmatrix}$ is called the $\sigma_{k,k}$-transformation of $\mathbf{M}$.

Now we modify the last round of `Rescue-XLIX-Prime` preimage polynomial systems analog to [Ste24b, Theorem 3.8].

Theorem 3.8. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}^n$ be such that $n = r_{in} + r_{out}$, and let $\mathcal{F}_{CICO} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq r} \subset \mathbb{F}_q \left[\mathbf{x}_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, \mathbf{x}_{out} \right]$ be parameters of a `Rescue-XLIX-Prime` CICO polynomial system with parameters d, n, r, r_{in} and r_{out}. Let $>_{\mathbf{W}, LEX}$ be the weight order from Theorem 3.4. Assume that* *(i)* $n > 2$,

(ii) $r_{in} < n - 1$,

(iii) $\operatorname{rank}\left(\rho_{r_{in},r_{in}}(\mathbf{M}_0)\right) = r_{in}$,

(iv) $\mathbf{M}_i$ *is in non-singular* $\rho_{r_{in},r_{in}}$*-position for all* $1 \leq i \leq 2 \cdot r - 1$,

(v) *for* $\mathbf{M}_{2\cdot r}$*:*

  (a) $\operatorname{rank}\left(\tau_{r_{out},r_{out}}\left(\mathbf{M}_{2\cdot r}^{-1}\right)\right) = r_{out}$, *and*

  (b) *let* $\mathbf{A}_{2\cdot r} \in \mathbb{F}_q^{r_{in}\times r_{out}}$ *and* $\mathbf{N}_{2\cdot r} \in \mathbb{F}_q^{r_{out}\times r_{out}}$ *be the matrices of the* $\tau_{r_{out},r_{out}}$ *transformation of* $\mathbf{M}_{2\cdot r}^{-1}$*, then* $\mathbf{A}_{2\cdot r}\mathbf{N}_{2\cdot r}$ *has at least two non-zero entries on every row.*

Then

(1) *A* $>_{\mathbf{w},LEX}$*-Gröbner basis for* $\mathcal{F}_{CICO}$ *can be computed via linear transformations.*

(2) $\dim_{\mathbb{F}_q}\left(\mathcal{F}_{pre}\right) = d^{n\cdot r}$.

Proof. Let $\mathbf{g}^{(0)}, \dots, \mathbf{g}^{(2\cdot r)}$ be as in Theorem 3.4, we introduce a modification for the last round

$$\begin{aligned}\hat{\mathbf{g}}^{(2\cdot r)} &= \begin{pmatrix} \mathbf{I}_{r_{in}\times r_{in}} & -\mathbf{A}_{2\cdot r} \\ \mathbf{0}_{r_{out}\times r_{in}} & \mathbf{I}_{r_{out}\times r_{out}} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{r_{in}\times r_{in}} & \mathbf{0}_{r_{in}\times r_{out}} \\ \mathbf{0}_{r_{out}\times r_{in}} & \mathbf{N}_{2\cdot r} \end{pmatrix} \mathbf{M}_{2\cdot r}^{-1}\mathbf{g}^{(2\cdot r)} \\ &= \begin{pmatrix} \mathbf{I}_{r_{in}\times r_{in}} & -\mathbf{A}_{2\cdot r}\mathbf{N}_{2\cdot r} \\ \mathbf{0}_{r_{out}\times r_{out}} & \mathbf{N}_{2\cdot r} \end{pmatrix} \mathbf{x}^{(2\cdot r)} + \hat{\mathbf{c}}_{2\cdot r} - \begin{pmatrix} \mathbf{0}_{r_{in}\times r_{in}} & \mathbf{B}_{2\cdot r} - \mathbf{A}_{2\cdot r}\mathbf{C}_{2\cdot r} \\ \mathbf{I}_{r_{out}\times r_{out}} & \mathbf{C}_{2\cdot r} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{out} \\ \boldsymbol{\beta} \end{pmatrix},\end{aligned}$$

which is possible due to $\operatorname{rank}\left(\tau_{r_{out},r_{out}}\left(\mathbf{M}_{2\cdot r}^{-1}\right)\right) = r_{out}$. We claim that

$$\mathcal{G} = \left\{\mathbf{g}^{(i)}\right\}_{0\leq i\leq r-1} \cup \left\{\hat{\mathbf{g}}^{(2\cdot r)}\right\}$$

is the $>_{\mathbf{w},LEX}$-Gröbner basis.

- For $0 \leq i \leq r - 1$, computation of the leading monomials is identical to Theorem 3.4.
- For $i = r$, due to the assumption that $\mathbf{A}_{2\cdot r}\mathbf{N}_{2\cdot r}$ has two non-zero entries on every row, the weights $\mathbf{w}_0, \dots, \mathbf{w}_{r-1}$ produce ties for $\mathbf{g}^{(2\cdot r)}|^{r_{in}}$.

  Also, the weights $\mathbf{w}_0, \dots, \mathbf{w}_{r-1}$ produce ties on $\mathbf{g}^{(2\cdot r)}|_{r_{out}}$ since $\mathbf{N}_{2\cdot r}$ is an invertible matrix.

  Therefore, we have to decide via $\mathbf{w}_{2\cdot r}$ which yields

$$\mathrm{LM}_{>_{\mathbf{w},LEX}}\left(\hat{\mathbf{g}}^{(2\cdot r)}\right) = \begin{pmatrix} \mathbf{x}^{(2\cdot r)}\big|^{r_{in}} \\ \mathbf{x}_{out} \end{pmatrix}.$$

Pairwise coprime leading monomials implies being a Gröbner basis by Lemma 2.9.

Counting the number of monomials not contained in the ideal of leading terms is analog to Theorem 3.4. □

Again, this proof fails if $n > 2$ and $r_{in} = 1$, since $\mathbf{A}_{2\cdot r}\mathbf{N}_{2\cdot r}$ is a column vector. Though, we can again fix this by considering the weights from Proposition 3.5.

Proposition 3.9. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}$ be integers such that $n = r_{in} + r_{out}$, and let $\mathcal{F}_{CICO} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r} \subset \mathbb{F}_q \left[\mathbf{x}_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, \mathbf{x}_{out} \right]$ be a* `Rescue-XLIX-Prime` *preimage polynomial system with the parameters d, n, r, r_{in} and r_{out}. Let $>_{\mathbf{W}, LEX}$ be the weight order from Proposition 3.5. Assume that*

(i) $n > 2$,

(ii) $r_{in} = n - 1$,

(iii) for all $1 \leq i \leq 2 \cdot r - 1$*:*

(a) $\operatorname{rank}\left(\rho_{r_{in}, r_{in}} (\mathbf{M}_i) \right) = r_{in}$,

(b) let $\mathbf{N}_i \in \mathbb{F}_q^{r_{in} \times r_{in}}$ be the matrix of the $\rho_{r_{in}, r_{in}}$-transformation of $\mathbf{M}_i$, then $\mathbf{N}_i$ has at least two non-zero entries in every row, and

(iv) $\operatorname{rank}\left(\tau_{r_{out}, r_{out}} \left(\mathbf{M}_{2 \cdot r}^{-1} \right) \right) = r_{out}$.

Then

(1) A $>_{\mathbf{W}, LEX}$-Gröbner basis for $\mathcal{F}_{pre}$ can be computed via linear transformations.

(2) $\dim_{\mathbb{F}_q} \left(\mathcal{F}_{CICO} \right) = d^{n \cdot r}$.

Proof. Let $\mathcal{G}$ as in Theorem 3.8, we claim that this is the $>_{\mathbf{W}, LEX}$-Gröbner basis.

- For $0 \leq i \leq 2 \cdot r - 1$, computation of the leading monomials is analog to Proposition 3.5.
- For $i = r$, let $0 \leq j \leq 2 \cdot r - 2$, if the matrix $\mathbf{A}_{2 \cdot r} \mathbf{N}_{2 \cdot r}$ has a zero row, then we have a trivial decision for a term of $\mathbf{x}^{(2 \cdot r)} |^{r_{in}}$ on that row of $\mathbf{g}^{(2 \cdot r)} |^{r_{in}}$. Otherwise, we have two terms present of weight 1, so we have a tie.

  In case of a tie, for $\mathbf{w}_{2 \cdot r - 1}$ all terms in $\mathbf{g}^{(2 \cdot r)} |^{r_{in}}$ have weight 0, so we have a trivial tie.

  Finally, decision by $\mathbf{w}_{2 \cdot r}$ yields

  $$\operatorname{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \Big|^{r_{in}} \right) = \mathbf{x}^{(2 \cdot r)} \Big|^{r_{in}}.$$

  Analog for $\mathbf{g}^{(2 \cdot r)} |_{r_{out}}$, the weights $\mathbf{w}_0, \dots, \mathbf{w}_{2 \cdot r - 2}$ produce ties since $\mathbf{N}_{2 \cdot r}$ is invertible, but for $\mathbf{w}_{2 \cdot r - 1}$ the terms $\mathcal{S} \left(\mathbf{x}^{(2 \cdot r)} \right)$ have weight 0 and the ones of $\mathbf{x}_{out}$ have weight 1, so

  $$\operatorname{LM}_{>_{\mathbf{W}, LEX}} \left(\mathbf{g}^{(2 \cdot r)} \Big|_{r_{out}} \right) = \mathbf{x}_{out}.$$

So, being a Gröbner basis follows from Lemma 2.9

Counting the number of monomials not contained in the ideal of leading terms is analog to Proposition 3.5. □

For $n = 2$ the previous proofs again fail, but via a modification analog to Corollary 3.6 we can again fix this.

Corollary 3.10. *Let $\mathbb{F}_q$ be a finite field, let $d, n, r, r_{in}, r_{out} \in \mathbb{Z}_{\geq 1}$ be integers such that $n = 2$, $r_{in} = 1$ and $n = r_{in} + r_{out}$, and let $\mathcal{F}_{CICO} = \left\{ \mathbf{f}_{pre}^{(i)} \right\}_{0 \leq i \leq 2 \cdot r} \subset \mathbb{F}_q \left[x_{in}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2 \cdot r)}, x_{out} \right]$ be a* `Rescue-XLIX-Prime` *CICO polynomial system with the parameters d, n, r, r_{in} and r_{out}. Let $>_{\mathbf{W}, LEX}$ be the weight order from Corollary 3.6. Assume that*

(i) $\mathbf{M}_i$ is in upper non-singular $\rho_{r_{in}, r_{in}}$-position for all $0 \leq i \leq 2 \cdot r - 1$, and

(ii) $\operatorname{rank}\left(\tau_{r_{out},r_{out}}\left(\mathbf{M}_{2\cdot r}^{-1}\right)\right) = r_{out}$.

Then

(1) A $>_{\mathbf{W},LEX}$*-Gröbner basis for* $\mathcal{F}_{CICO}$ *can be computed via linear transformations.*

(2) $\dim_{\mathbb{F}_q}\left(\mathcal{F}_{pre}\right) = d^{2\cdot r}$.

Proof. Let $\mathbf{g}^{(0)}, \dots, \mathbf{g}^{(2\cdot r)}$ be as in Theorem 3.4, we modify the polynomials in the last round

$$\begin{aligned}\hat{\mathbf{g}}^{(2\cdot r)} &= \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -A_{2\cdot r} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & N_{2\cdot r} \end{pmatrix} \mathbf{M}_{2\cdot r}^{-1} \mathbf{g}^{(2\cdot r)} \\ &= \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & -A_{2\cdot r} \cdot N_{2\cdot r} \\ 0 & N_{2\cdot r} \end{pmatrix} \mathbf{x}^{(2\cdot r)} + \hat{\mathbf{c}}_{2\cdot r} - \begin{pmatrix} 0 & B_{2\cdot r} - A_{2\cdot r} \cdot C_{2\cdot r} \\ 1 & C_{2\cdot r} \end{pmatrix} \begin{pmatrix} x_{out} \\ \beta \end{pmatrix} \right),\end{aligned}$$

where $\gamma \in \mathbb{F}_q^\times$ is chosen such that the coefficient of $x_2^{(2\cdot r)}$ in $\hat{g}_1^{(2\cdot r)}$ is non-zero. Then, $x_1^{(2\cdot r)}$, $x_2^{(2\cdot r)}$ and x_{out} are present in $g_1^{(2\cdot r)}$, but only $x_2^{(2\cdot r)}$ and x_{out} are present in $g_2^{(2\cdot r)}$. Proving that

$$\mathcal{G} = \left\{\mathbf{g}^{(i)}\right\}_{0 \leq i \leq r-1} \cup \left\{\hat{\mathbf{g}}^{(2\cdot r)}\right\}$$

is the $>_{\mathbf{W},LEX}$-Gröbner basis of $\mathcal{F}_{\text{pre}}$ is then identical to Corollary 3.6. □

4 Cryptanalytic Applications

As outlined in [Ste24b, §4], with our `Rescue-XLIX` Gröbner bases we can now either perform term order conversion to LEX or compute the eigenvalues of the multiplication matrices to compute the variety of the `Rescue-XLIX` polynomial system.

Let $I \subset P = K[x_1, \dots, x_n]$ be a zero-dimensional ideal, and let $D = \dim_K(I)$ be the K-vector space dimension of the quotient ring P/I. In addition, we denote the variety of I as

$$\mathcal{V}(I) = \{\mathbf{x} \in K^n \mid \forall f \in I \colon f(\mathbf{x}) = 0\}. \tag{14}$$

Term order conversion via the original FGLM algorithm [FGLM93] can be performed in $\mathcal{O}\left(n \cdot D^3\right)$, but an improved probabilistic variant [FGHR14] achieves $\mathcal{O}\left(n \cdot D^\omega\right)$, where $2 \leq \omega < 2.37286$ [AW21] is a linear-algebra constant, and an improved sparse linear algebra variant [FM17] achieves $\mathcal{O}\left(\sqrt{n} \cdot D^{2+\frac{n-1}{n}}\right)$. Note that full complexity analysis in [FGHR14, FM17] was only done for the DRL term order.

Over a finite field $K = \mathbb{F}_q$, factorization of the univariate LEX polynomial can be speed-up by computing its greatest common divisor (GCD) with the field equation $x^q - x$. This has the convenient benefit, that all remaining roots of the GCD come from $\mathbb{F}_q$ and have multiplicity 1. According to Bariant et al. [BBLP22, §3.1] the GCD can be computed in

$$\mathcal{O}\Big(D \cdot \log(D) \cdot \log\big(\log(D)\big) \cdot \big(\log(D) + \log(q)\big)\Big), \tag{15}$$

for $D \leq q$, else D and q have to be exchanged in the complexity estimate.

On the other hand, LEX term order conversion can be bypassed via linear algebra-based techniques. Kreuzer & Robbiano [KR16, Chapter 6] discuss how the variety $\mathcal{V}(I)$ can be computed if a K-vector space basis $\mathcal{B}$ of P/I is known. In case one knows a $>$-Gröbner basis of I, then $\mathcal{B}$ consists of the monomials not contained in the ideal of leading terms, and for zero-dimensional ideals this basis is always finite [KR00, Proposition 3.7.1]. Let $f \in P$, the multiplication map for f in P/I, see [KR16, Definition 4.1.4] is defined as

$$\theta_f : P/I \to P/I, \qquad x \mapsto f \cdot x. \tag{16}$$

Since P/I is a finite dimensional K-vector space and the map is K-linear, θ_f can be represented as matrix. This matrix is called the multiplication matrix $\mathbf{M}_f$ of f in R/I. Via a $>$-Gröbner basis $\mathcal{G} \subset I$ one computes the multiplication matrix as follows: Index the columns of $\mathbf{M}_f$ by the elements of $\mathcal{B}$, and rows by $f \cdot b$, where $b \in \mathcal{B}$. Next compute $f \cdot b \mod \mathcal{G}$, extract its coefficient vector with respect to $\mathcal{B}$ and fill it into the row $b \cdot f$. Over the algebraic closure $\bar{K}$ and for $f = x_i$, the i^{th} coordinate of a point $\mathbf{x} \in \mathcal{V}_{\bar{K}}(I)$ is an eigenvalue of the multiplication matrix $\mathbf{M}_{x_i}$, see [KR16, Corollary 6.2.3]. Therefore, $\mathcal{V}_{\bar{K}}(I)$ can be computed via the eigenvalues of $\mathbf{M}_{x_1}, \dots, \mathbf{M}_{x_n}$, taking all possible combinations of the eigenvalues, and finally verifying whether a combination is indeed a point in the variety. This approach is known as the *Eigenvalue Method* [KR16, Algorithm 6.2.7]. The eigenpolynomial is computed via the determinant, and with fast matrix multiplication the determinant of a matrix $\mathbf{M} \in K^{N \times N}$ can be computed in, see [AHU74, Theorem 6.6], $\mathcal{O}(N^\omega)$, where again $2 \leq \omega < 2.37286$. Hence, the Eigenvalue Method has complexity

$$\mathcal{O}\left(n \cdot D^\omega\right), \tag{17}$$

which is identical probabilistic FGLM algorithm [FGHR14].

For Rescue-XLIX polynomial systems this estimate can be slightly improved. For the cipher, we only care about the solutions of the master key, i.e. we only need to compute n eigenpolynomials instead of the generic $n \cdot (4 \cdot r + 1)$, so Equation (17) improves to

$$\mathcal{O}\left(n \cdot d^{\omega \cdot n \cdot r}\right) \tag{18}$$

for affine key schedules, and

$$\mathcal{O}\left(n \cdot d^{2 \cdot \omega \cdot n \cdot r}\right) \tag{19}$$

for non-affine key schedules. Since Rescue-XLIX is a permutation, for a sponge function we only need to compute the eigenpolynomials for the input or the output variables instead of the generic $n \cdot (2 \cdot r + 1)$, so Equation (17) improves to

$$\mathcal{O}\left(\min\{r_{in}, r_{out}\} \cdot d^{\omega \cdot n \cdot r}\right). \tag{20}$$

Lastly, we only care about $\mathbb{F}_q$-valued solutions, so we will always opt for the GCD method to extract the cryptographically relevant solutions whose complexity is given by Equation (15).

In [Ste24b, §4] it was pointed out that the sparsity of the Gröbner bases could be exploited to construct an eigenpolynomial faster than the generic $\mathcal{O}(D^\omega)$. Following this rational, for a given security level κ we say that an instance resists generic eigenpolynomial construction if

$$\log_2\left(\min\{r_{in}, r_{out}\}\right) + \omega \cdot \log_2(D) \geq \kappa, \tag{21}$$

and we say the instance resists root extraction if

$$\begin{rcases} \begin{cases} D \cdot \log(D) \cdot \log\left(\log(D)\right) \cdot \left(\log(D) + \log(q)\right), & D \leq q \\ q \cdot \log(q) \cdot \log\left(\log(q)\right) \cdot \left(\log(q) + \log(D)\right), & D > q \end{cases} \end{rcases} \geq 2^\kappa. \tag{22}$$

I.e., for root extraction it is assumed that eigenpolynomial construction is either for free or can be done below the root extraction complexity.

4.1 Rescue-XLIX Complexity Estimation

In this section we assume that a Rescue-XLIX instance satisfies the necessary conditions for the Gröbner bases. Recall that for the cipher polynomial system, see Theorem 3.1, the $\mathbb{F}_q$-vector space dimension is given by

$$D_{\texttt{Rescue-XLIX}} = \begin{cases} d^{n \cdot r}, & \text{affine key schedule}, \\ d^{2 \cdot n \cdot r}, & \text{non-affine key schedule}, \end{cases} \tag{23}$$

and that the $\mathbb{F}_q$-vector space dimension of a fully determined `Rescue-XLIX-Prime` preimage or CICO polynomial system is given by

$$D_{\texttt{Rescue-XLIX-Prime}} = d^{n \cdot r}. \tag{24}$$

In case one is given an underdetermined preimage/CICO polynomial system, i.e. $r_{in} + r_{out} > n$, one has to guess input/output variables until $n = \tilde{r}_{in} + \tilde{r}_{out}$. In case one is given an overdetermined preimage/CICO polynomial system, i.e. $r_{in} + r_{out} < n$, one either has to forget input/output constants and treat them as variables until $n = \tilde{r}_{in} + \tilde{r}_{out}$, or one has to recompute the Gröbner basis. We note that the latter approach is beyond the scope of this paper. In Table 1 we provide sample complexities for `Rescue-XLIX-Prime`. The round numbers have been computed via the `SageMath` [Sag23] round numbers tool for `Rescue`.[3] For all instances a security level of 128 bits was used. All parameter sets achieve at least 128 bits of security against eigenpolynomial construction, but no instance achieves 128 bits of security against root extraction. Observe that the complexities of Table 1 can also be used for the `Rescue-XLIX` cipher with affine key schedule.

Table 1: `Rescue-XLIX-Prime` complexity estimations for generic eigenpolynomial construction and $\mathbb{F}_q$-valued root extraction. All estimations use $\omega = 2$, c denotes the capacity of the sponge function.

$\log_2(q)$	n	c	d	r	Eigenpolynomial (bits)	Root extraction (bits)
31	12	8	5	8	446	45
31	16	8	5	8	595	45
64	8	4	7	8	360	79
64	12	4	7	8	540	80
256	2	1	5	20	186	109
256	3	1	5	14	196	114
256	4	1	5	11	205	119

4.1.1 Ethereum Challenge

In 2021 Ethereum foundation hosted a CICO cryptanalysis challenge [Eth21] for various AO hash functions among them `Rescue-XLIX-Prime`. For the challenge one had to solve the CICO problems

$$\texttt{Rescue-XLIX-Prime}\begin{pmatrix} x_{in,1} \\ 0 \end{pmatrix} = \begin{pmatrix} x_{out,1} \\ 0 \end{pmatrix}, \tag{25}$$

$$\texttt{Rescue-XLIX-Prime}\begin{pmatrix} x_{in,1} \\ x_{in,2} \\ 0 \end{pmatrix} = \begin{pmatrix} x_{out,1} \\ x_{out,2} \\ 0 \end{pmatrix} \tag{26}$$

over the prime $p = 18446744073709551557$ with $d = 3$ for various round numbers. For three branches the polynomial system is not fully determined, hence we have to guess one variable. We note that Bariant et al. [BBLP22, §4.4] also investigated the challenge for `Rescue-XLIX-Prime`. For our Gröbner bases we present complexities for an attack on `Rescue-XLIX-Prime` in Table 2. For the full model, all parameter sets achieve the claimed security level for eigenpolynomial construction.

Bariant et al. found a trick [BBLP22, §4.2] to bypass the first two SPN rounds of `Rescue-XLIX-Prime` when $n = 3$. By their analysis, one can consider the input state of

[3] https://github.com/KULeuven-COSIC/Marvellous

the second full round to be of the form

$$\mathbf{x}^{(3)} = \begin{pmatrix} a_1 \cdot x \\ a_2 \cdot x \\ b \end{pmatrix} = \begin{pmatrix} a_1 & 0 & 0 \\ a_2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ 0 \\ b \end{pmatrix}, \tag{27}$$

where x is a new variable and $a_1, a_2, b \in \mathbb{F}_q$ are such that $a_1 \cdot a_2 \neq 0$. Our `Rescue-XLIX-Prime` Gröbner basis from Theorem 3.8 can be extended to this trick, since one just cuts off one full round and formulates a smaller CICO problem. In particular, the quotient space dimension becomes

$$\hat{D}_{\texttt{Rescue-XLIX-Prime}} = \dim_{\mathbb{F}_q} \left(\hat{\mathcal{F}}_{\text{CICO}} \right) = d^{3 \cdot (r-1)}. \tag{28}$$

For the bypassed rounds, we also provide the complexities of a Gröbner basis attack on the `Rescue-XLIX-Prime` challenge in Table 2. With the bypassed rounds, the parameter sets with $n = 3$ do not achieve the claimed security level for eigenpolynomial construction as well as root extraction.

Table 2: `Rescue-XLIX-Prime` Ethereum challenge [Eth21] complexity estimation. The challenge is defined over the prime $p = 18446744073709551557$ and $d = 3$. All estimations use $\omega = 2$.

n	r	Eigenpolynomial (bits)	Root extraction (bits)	Security level (bits)
		Full model		
3	4	39	30	37.5
2	6	39	30	37.5
2	7	45	34	43.5
3	5	48	36	45
2	8	51	37	49.5
		First two rounds bypassed		
3	4	29	25	37.5
3	5	39	30	45

5 Discussion

With the works of [Ste24a, Ste24b] we can compare the $\mathbb{F}_q$-vector space dimension of SPN-based AO primitives, see Table 3. First we notice that the SPN cipher and `Rescue-XLIX` with affine key schedule have the same dimension, although `Rescue-XLIX` requires $2 \cdot r$ SPN evaluations. But for the SPN sponge and Poseidon the dimension depends on the rate r_{in}. In particular, an adversary has control over r_{in} since he can guess additional variables on the input and ignore constants on the output until $\tilde{r}_{in} = 1$. On the other hand, for `Rescue-XLIX-Prime` the dimension is always independent of the rate r_{in}, and it coincides with the one for the affine key schedule.

Table 3: Comparison of $\mathbb{F}_q$-vector space dimensions for the SPN, Hades, Poseidon and `Rescue-XLIX`. With n we denote the number of branches, d the degree of the power permutation, r the number of rounds of the SPN and `Rescue-XLIX`, r_f and r_p the number of full/partial rounds of Poseidon and Hades, and r_{in} denotes the input rate of a sponge polynomial system.

	$\mathbb{F}_q$-vector space dimension		
Primitive	Cipher	Sponge	Reference
SPN	$d^{n\cdot r}$	$d^{r_{in}\cdot r}$	[Ste24a, Theorem 6.2]
Hades	$d^{2\cdot n\cdot r_f+r_p}$	n.a.	[Ste24a, Theorem 6.2]
Poseidon	n.a.	$d^{2\cdot r_{in}\cdot r_f+r_p}$	[Ste24b, §3.2, B.2]
`Rescue-XLIX`	$\begin{cases} d^{n\cdot r}, & \text{affine key schedule,} \\ d^{2\cdot n\cdot r}, & \text{non-affine key schedule} \end{cases}$	$d^{n\cdot r}$	Section 3

The original `Rescue` Gröbner basis cryptanalysis [AAB+20, §6.1] relied on extrapolation of small scale experiments, with our Gröbner bases this analysis becomes superfluous since every Gröbner basis is equally capable to compute the variety via the Eigenvalue Method.

As discussed in [Ste24b, §5], any other polynomial model for `Rescue-XLIX` that is contained in one of our iterated models (Definition 2.4) can be ignored. The number of points in the variety as well as the vector space dimension of the other model is always at least as big as the ones for the iterated model. Hence, solving for another model is at least as difficult as solving for the iterated model (as long as it is contained in the iterated model).

Acknowledgments

Matthias Steiner has been supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 725042) and the enCRYPTON project (grant agreement No. 101079319).

References

[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symm. Cryptol.*, 2020(3):1–45, 2020. doi:10.13154/tosc.v2020.i3.1-45.

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, Heidelberg, September 2019. doi:10.1007/978-3-030-29962-0_8.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_7.

[AHU74] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1974.

[AKM⁺22] Tomer Ashur, Al Kindi, Willi Meier, Alan Szepieniec, and Bobbin Threadbare. Rescue-prime optimized. Cryptology ePrint Archive, Report 2022/1577, 2022. https://eprint.iacr.org/2022/1577.

[AW21] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Dániel Marx, editor, *32nd SODA*, pages 522–539. ACM-SIAM, January 2021. doi:10.1137/1.9781611976465.32.

[BBC⁺23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions: Anemoi permutations and Jive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 507–539. Springer, Heidelberg, August 2023. doi:10.1007/978-3-031-38548-3_17.

[BBLP22] Augustin Bariant, Clémence Bouvier, Gaëtan Leurent, and Léo Perrin. Algebraic attacks against some arithmetization-oriented primitives. *IACR Trans. Symm. Cryptol.*, 2022(3):73–101, 2022. doi:10.46586/tosc.v2022.i3.73-101.

[BCL⁺20] Tim Beyne, Anne Canteaut, Gregor Leander, María Naya-Plasencia, Léo Perrin, and Friedrich Wiemer. On the security of the rescue hash function. Cryptology ePrint Archive, Report 2020/820, 2020. https://eprint.iacr.org/2020/820.

[BCP23] Aurélien Boeuf, Anne Canteaut, and Léo Perrin. Propagation of subspaces in primitives with monomial sboxes: Applications to Rescue and variants of the AES. *IACR Trans. Symm. Cryptol.*, 2023(4):270–298, Dec. 2023. doi:10.46586/tosc.v2023.i4.270-298.

[BDND⁺21] Mina Bigdeli, Emanuela De Negri, Manuela Muzika Dizdarevic, Elisa Gorla, Romy Minko, and Sulamithe Tsakou. Semi-regular sequences and other random systems of equations. In Alina Carmen Cojocaru, Sorina Ionica, and Elisa Lorenzo García, editors, *Women in Numbers Europe III: Research Directions in Number Theory*, pages 75–114, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-77700-5_3.

[BDPA13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 313–314. Springer, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_19.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. Ecrypt Hash Workshop, 2007. URL: https://keccak.team/files/SpongeFunctions.pdf.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, Heidelberg, April 2008. doi:10.1007/978-3-540-78967-3_11.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions. NIST SHA-3 competition (round 3), 2011. URL: https://keccak.team/files/CSF-0.1.pdf.

[BFS04] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proceedings of the International Conference on Polynomial System Solving*, pages 71–74, 2004.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Universität Innsbruck, 1965.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer International Publishing, 4 edition, 2015. doi:10.1007/978-3-319-16721-3.

[Eth21] The Ethereum Foundation. ZK Hash Function Cryptanalysis Bounties 2021. https://www.zkhashbounties.info/, 2021. Accessed: 2024-03-18.

[FGHR14] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Sub-cubic change of ordering for Gröbner basis: A probabilistic approach. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, page 170–177, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2608628.2608669.

[FGLM93] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993. doi:10.1006/jsco.1993.1051.

[FM17] Jean-Charles Faugère and Chenqi Mou. Sparse FGLM algorithms. *J. Symb. Comput.*, 80:538–569, 2017. doi:10.1016/j.jsc.2016.07.025.

[FP19] Jean-Charles Faugère and Ludovic Perret. Algebraic attacks against `stark`-friendly ciphers. Appearing as Appendix A in https://eprint.iacr.org/2020/948, 2019. Version 1.2.

[GHR+23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst meets fluid-SPN: Griffin for zero-knowledge applications. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 573–606. Springer, Heidelberg, August 2023. doi:10.1007/978-3-031-38548-3_19.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 519–535. USENIX Association, August 2021.

[GKS23] Lorenzo Grassi, Dmitry Khovratovich, and Markus Schofnegger. Poseidon2: A faster version of the poseidon hash function. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 23*, volume 14064 of *LNCS*, pages 177–203. Springer Nature, July 2023. doi:10.1007/978-3-031-37679-5_8.

[GLR$^+$20] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 674–704. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_23.

[Jea16] Jérémy Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[KR00] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1 edition, 2000. doi:10.1007/978-3-540-70628-1.

[KR05] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 2*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1 edition, 2005. doi:10.1007/3-540-28296-3.

[KR16] Martin Kreuzer and Lorenzo Robbiano. *Computational Linear and Commutative Algebra*. Springer International Publishing, Cham, 1 edition, 2016. doi:10.1007/978-3-319-43601-2.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite fields*. Encyclopedia of mathematics and its applications. Cambridge Univ. Press, Cambridge, 2 edition, 1997.

[Rob86] Lorenzo Robbiano. On the theory of graded structures. *J. Symb. Computat.*, 2(2):139–170, 1986. doi:10.1016/S0747-7171(86)80019-0.

[SAD20] Alan Szepieniec, Tomer Ashur, and Siemen Dhooghe. Rescue-prime: a standard specification (SoK). Cryptology ePrint Archive, Report 2020/1143, 2020. https://eprint.iacr.org/2020/1143.

[Sag23] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.2)*, 2023. https://www.sagemath.org.

[Ste24a] Matthias Johann Steiner. Solving degree bounds for iterated polynomial systems. *IACR Trans. Symm. Cryptol.*, 2024(1):357–411, Mar. 2024. doi:10.46586/tosc.v2024.i1.357-411.

[Ste24b] Matthias Johann Steiner. A zero-dimensional Gröbner basis for Poseidon. Cryptology ePrint Archive, Paper 2024/310, 2024. URL: https://eprint.iacr.org/2024/310.